

Vizualizace monitorování provozně technických funkcí v Inteligentní budově v rámci IoT

Visualization of Operational and Technical Functions in Intelligent Building

Bc. Karel Fajt

Diplomová práce

Vedoucí práce: doc. Ing. Jan Vaňuš, Ph.D.

Ostrava, 2021

Abstrakt

Cílem této diplomové práce je zajištění výpočtu predikce koncentrace oxidu uhličitého z dat PI System. Prvním krokem je prozkoumání možností, které nabízí v rámci analýzy dat a strojového učení cloudové platformy IBM Cloud a Microsoft Azure. Poznatky, získané při práci s cloudovými nástroji, jsou sepsány do vlastní kapitoly. V další části práce jsou vypracovány experimenty. Jejich výsledky vedou ke zvolení vhodné predikční metody a jejího nastavení pro potřeby této práce. Prostřednictvím PI Web API je vytvořena konektivita pro čtení a zápis dat z PI Server. V jazyce Python je sepsán kód, který slouží jak pro dlouhodobý výpočet predikce CO₂, tak pro komunikaci s PI Web API. Dále je v rámci streamování dat v reálném čase popsána práce s nástrojem PI Integrator for Business Analytics a návaznost na Apache Kafka. Na závěr je vytvořena vizualizace provozně technických funkcí dvou bytů v nástroji PI Vision. Vizualizace obsahuje i data predikovaných hodnot CO₂.

Klíčová slova

OSIsoft, PI System, PI Web API, IBM Cloud, Watson studio, Microsoft Azure, Machine Learning studio, neuronové sítě, náhodný les, CHAID, predikce, CO₂, PI Integrator for Business Analytics

Abstract

The aim of this dissertation is to ensure the calculation of the prediction of carbon dioxide concentration from PI System data. The first step is to explore the possibilities that IBM Cloud and Microsoft Azure offer in data analysis and machine learning. The knowledge gained when working with cloud tools is written up in a separate chapter. In the next part of the paper, experiments are developed. Their results lead to the selection of a suitable prediction method and its setting for the needs of this work. Connectivity for reading and writing data from the PI Server is created via the PI Web API Code that is used both for the long-term calculation of CO₂ prediction and for communication with PI Web API is written in Python. Furthermore, the work with the PI Integrator for the Business Analytics tool and the connection to Apache Kafka are described within real-time data streaming. Finally, a visualization of the operational and technical functions of the two bytes in the PI Vision tool is created. The visualization also contains data of predicted CO₂ values.

Keywords

OSIsoft, PI System, PI Web API, IBM Cloud, IBM Watson studio, Microsoft Azure, Neural Networks, Random Forest, CHAID, CO₂, PI Integrator for Business Analytics

Poděkování

Rád bych poděkoval svému vedoucímu panu doc. Ing. Janu Vaňušovi, PhD. za odborné konzultace, čas a prostředky, díky kterým mohla tato diplomová práce vzniknout. Dále bych rád poděkoval panu Ing. Ondřeji Kafnovi za jeho rady a pomoc s PI System.

Obsah

1	Úvod	12
2	Rešerše současného stavu vizualizace provozně technických funkcí v Inteligentní budově	13
3	PI System	17
3.1	Datová část	17
3.2	PI Server.....	18
3.3	PI Web API	19
3.4	PI Integrator for Business Analytics.....	20
3.5	PI Vision	21
3.6	PI ProcessBook	22
3.7	Způsoby uložení dat	23
3.8	PI System Security	23
4	Strojové učení.....	25
4.1	Hluboké učení.....	25
4.2	Popis modelů strojového učení použitých v této práci	26
4.2.1	Neuronové sítě	26
4.2.2	Náhodný les.....	30
4.2.3	CHAID.....	32
4.3	Kritéria přesnosti predikce	33
4.3.1	Linear Correlation (LC).....	34
5	Analýza dat v rámci IoT	36
5.1	IBM Cloud	36
5.1.1	SPSS Modeler.....	36
5.1.2	AutoAI.....	38
5.1.3	Notebook.....	39
5.1.4	Nasazení modelů	40
5.1.5	Cenová politika cloudu	40
5.2	Microsoft Azure	41

5.2.1	Designer.....	41
5.2.2	Automated ML.....	42
5.2.3	Notebook.....	43
5.2.4	Nasazení modelů	44
5.2.5	Cenová politika cloudu	44
5.3	Zhodnocení práce s cloudy.....	44
6	Experimenty	45
6.1	Výběr metod.....	45
6.2	Experimenty	45
6.3	Zhodnocení experimentů	56
7	Výpočet predikce v reálném čase.....	57
7.1	Konektivita dat, PI Web API.....	57
7.2	Výpočet predikce, predikční model.....	63
7.3	PI Integrator for Business Analytics, Apache Kafka.....	66
8	Vizualizace	69
8.1	Popis vizualizace	69
8.2	Vytvoření databáze pro vizualizaci	71
8.3	Konverze datových typů	72
8.4	Vytvořená vizualizace	74
9	Závěr	78
10	Literatura	80
11	Seznam přílohy	85

Seznam použitých symbolů a zkratek

SH	SmartHome
BACnet	Building Automation and Control Networks
CPIT TL	Centrum pokročilých inovačních technologií, těžké laboratoře
IoT	Internet of Things
JSON	JavaScript Object Notation
API	Application Programm ing Interface
PI SDK	PI Software Development Kit
PI AF	PI Asset Framework
PI SMT	PI Software Management Tools
REST	Representational State Transfer
CART	Classification and Regression Tree
NN	Neural Networks
RF	Random Forest
CHAID	Chi-squared Automated Interaction Detection
LC	Linear Corelation
MAE	Mean Absolut Error
MSE	Mean Square Error
MAPE	Mean Absolute Percentile Error
NP	Nadzemní Patro
HVAC	Heating, Ventilation and Air Conditioning
ppm	parts per million - počet molekul CO ₂ na milion molekul vzduchu

Seznam ilustrací

Obr. 1 Architektura PI System [28].....	17
Obr. 2 Architektura nasazení PI Web API.....	19
Obr. 3 Popis rozmístění prvků pro vytváření displeje v PI Vision [32]	22
Obr. 4 Znáznornění zabezpečení připojení do PI Data Archive a PI Asset Framework [37].....	23
Obr. 5 Návaznost oblastí umělé inteligence [38]	25
Obr. 6 Model neuronu [39]	26
Obr. 7 Aktivační přenosové funkce [39].....	28
Obr. 8 Uspořádání neuronů do vrstev v dopředné neuronové síti [40].....	29
Obr. 9 Vytvoření pipeline pro predikci CO ₂ , nástroj SPSS Modeler.....	38
Obr. 10 Výsledek experimentu AutoAI.....	39
Obr. 11 Ukázka prostředí Notebook Python 3.7	40
Obr. 12 Pipeline vytvoření v nástroji Designer.....	42
Obr. 13 Výsledky experimentu v nástroji Automated ML.....	43
Obr. 14 Příklad části kódu v Notebook.....	43
Obr. 15 Zapojení pipeline v SPSS Modeler, model neuronové sítě.....	46
Obr. 16 Graf porovnání naměřených a predikovaných hodnot CO ₂ . Modelem náhodného lesu, použito pět rozhodovacích stromů. Datový soubor 8.2.2020 – 15.2.2020. Zelená svislá čára značí příchod osoby do místnosti, černá svislá čára značí odchod. Příchod $t_1 = 10.2.2020\ 08:50$, odchod $t_1 = 10.2.2020\ 9:40$, délka pobytu $\Delta t_1 = 00:50$. Příchod $t_2 = 11.2.2020\ 07:30$, odchod $t_2 = 11.2.2020\ 10:10$, délka pobytu $\Delta t_2 = 02:40$. Příchod $t_3 = 11.2.2020\ 12:00$, odchod $t_3 = 11.2.2020\ 12:50$, délka pobytu $\Delta t_3 = 00:50$. Příchod $t_4 = 11.2.2020\ 15:20$, odchod $t_4 = 11.2.2020\ 15:40$, délka pobytu $\Delta t_4 = 00:20$	47
Obr. 17 Graf porovnání naměřených a predikovaných hodnot CO ₂ . Model CHAID, použito šest vrstev rozhodovacího stromu. Datový soubor 8.2.2020 – 15.2.2020. Zelená svislá čára značí příchod osoby do místnosti, černá svislá čára značí odchod. Příchod $t_1 = 10.2.2020\ 08:50$, odchod $t_1 = 10.2.2020\ 9:40$, délka pobytu $\Delta t_1 = 00:50$. Příchod $t_2 = 11.2.2020\ 07:30$, odchod $t_2 = 11.2.2020\ 10:10$, délka pobytu $\Delta t_2 = 02:40$. Příchod $t_3 = 11.2.2020\ 12:00$, odchod $t_3 = 11.2.2020\ 12:50$, délka pobytu $\Delta t_3 = 00:50$. Příchod $t_4 = 11.2.2020\ 15:20$, odchod $t_4 = 11.2.2020\ 15:40$, délka pobytu $\Delta t_4 = 00:20$	49
Obr. 18 Graf porovnání naměřených a predikovaných hodnot CO ₂ . Modelem neuronových sítí, použito 200 neuronů v první vnořené vrstvě, 100 neuronů ve vrstvě druhé. Datový soubor 8.2.2020 – 15.2.2020. Zelená svislá čára značí příchod osoby do místnosti, černá svislá čára značí odchod. Příchod $t_1 = 10.2.2020\ 08:50$, odchod $t_1 = 10.2.2020\ 9:40$, délka pobytu $\Delta t_1 = 00:50$. Příchod $t_2 = 11.2.2020\ 07:30$, odchod $t_2 = 11.2.2020\ 10:10$, délka pobytu $\Delta t_2 = 02:40$. Příchod $t_3 = 11.2.2020\ 12:00$, odchod $t_3 =$	

11.2.2020 12:50, délka pobytu $\Delta t_3 = 00:50$. Příchod $t_4 = 11.2.2020\ 15:20$, odchod $t_4 = 11.2.2020\ 15:40$, délka pobytu $\Delta t_4 = 00:20$.	51
Obr. 19 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden den 8.2.2015	52
Obr. 20 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden den 8.6.2015	52
Obr. 21 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden týden 8.2. – 15.2.2015	53
Obr. 22 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden týden 8.6. – 15.6.2015	54
Obr. 23 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden měsíc 1.2. – 28.2.2015	55
Obr. 24 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden měsíc 1.6. – 28.6.2015	55
Obr. 25 Blokový diagram konektivity mezi PI Serverem a PC	57
Obr. 26 Změna autentifikační metody na Basic	58
Obr. 27 Postup certifikátu z webového prohlížeče	58
Obr. 28 Povolení zápisu pro PI Web API, nástroj PI System Explorer	59
Obr. 29 Příklad nastavení oprávnění pro přístup k elementu v PI System Explorer	59
Obr. 30 Nastavení oprávnění čtení a zápisu pro PI Points, PI SMT	59
Obr. 31 Okaz pro vstup do helpu z PI Web API	60
Obr. 32 Ukázka volání requestu Batch v jazyku python	61
Obr. 33 Ukázka volání requestu pro zápis dat do PI Data Archive v jazyce Python	62
Obr. 34 Graf porovnání průběhů predikovaných a naměřených hodnot v reálném čase, PI Vision	65
Obr. 35 Blokový diagram komunikace mezi PI Server a koncovým uživatelem	66
Obr. 36 Nastavení Trigger pro Streaming View, PI Integrator for Business Analytics	67
Obr. 37 Ukázka spuštění serveru Zookeeper z příkazové řádky	67
Obr. 38 Seznam vytvořených Topics	67
Obr. 39 Vytvořený Target v PI Integrator pro Apache Kafka	68
Obr. 40 Výpis streamovaných dat z PI Integrator a Apache Kafka do okna příkazového řádku	68
Obr. 41 Budova CPIT TL3	69

Obr. 42 Blokové schéma rozvržení vizualizace	70
Obr. 43 Postup vytvoření nové databáze	71
Obr. 44 Postup vytvoření nového atributu prvku databáze	72
Obr. 45 Vytvoření tabulky Digital States	72
Obr. 46 Přetypování Pointu	73
Obr. 47 Úvodní obrazovka bytu na druhém patře	74
Obr. 48 Místnost 220, obývací pokoj s kuchyní	74
Obr. 49 Grafy pro místnost 220	75
Obr. 50 Grafy predikce koncentrace CO ₂ a příslušných kritérií	75
Obr. 51 Obrazovka vzduchotechnického zařízení na druhém patře	76
Obr. 52 Obrazovka provozně technických funkcí koupelny na druhém patře	76

Seznam tabulek

Tab. 1 Tabulka časové dotace služb IBM Cloud.....	41
Tab. 2 Ceny za použití virtuálních výkonů zařízení.....	41
Tab. 3 Ceny výpočetních výkonů virtuálního zařízení	44
Tab. 4 Hodnoty kritérií pro různá nastavení počtu vytvořených rozhodovacích stromů v modelu náhodného lesu.....	46
Tab. 5 Hodnoty kritérií pro různé délky intervalu vstupního datového souboru, model náhodného lesu	47
Tab. 6 Hodnoty kritérií pro různá nastavení maximální hloubky vytvořeného stromu modelu CHAID	48
Tab. 7 Hodnoty kritérií pro různé délky intervalu vstupního datového souboru, model CHAID	49
Tab. 8 Hodnoty kritérií pro různé počty neuronů ve vnořených vrstvách modelu neuronové sítě.....	50
Tab. 9 Hodnoty kritérií pro různé délky intervalu vstupního datového souboru, model neuronové sítě	51
Tab. 10 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho dne 8.2.2015....	53
Tab. 11 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho dne 8.6.2015....	53
Tab. 12 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho týdne 8.6. – 15.6.2015.....	54
Tab. 13 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho týdne 8.2. – 15.2.2015.....	54
Tab. 14 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho měsíce 1.2. – 28.2.2015.....	55
Tab. 15 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho měsíce 1.6. – 28.6.2015.....	56
Tab. 16 Výsledky kritérií predikčního modelu náhodný les v jazyce Python.....	63
Tab. 17 Porovná přesnosti predikce pro různé rozsahy hodnot CO ₂	64

1 Úvod

Cílů této práce je více. Hlavním cílem je zajištění výpočtu predikce koncentrace oxidu uhličitého z dat PI System. Jako prediktory jsou zde použity hodnoty vnitřní a venkovní teploty a relativní vlhkosti v místnosti. Sensory koncentrace CO₂ jsou finančně náročná zařízení, jejich použití zvyšuje pořizovací cenu instalace SmartHome (dále jenom SH). Motivací k zajištění výpočtu predikce koncentrace CO₂ je dosažení finančně úspornějšího systému bez nutnosti použití sensorů CO₂. Dalším cílem je vytvoření vizualizace provozně technických funkcí v nástroji PI Vision. Ta doplní stávající vizualizaci budovy FEI a dřevodomku. Posledním cílem je popis a praktické odzkoušení způsobů sdílení dat PI system v reálném čase za účelem aplikace modelů strojového učení na PI data. PI System je systém pro správu a ukládání velkého množství dat, sám však žádné nástroje pro tyto účely nenabízí.

V rámci teoretické části práce je nejprve provedena rešerše současného stavu, která ukazuje, jaké trendy jsou dnes používány v oblasti SH technologií. Dále je sepsána kapitola věnovaná popisu PI System. Jsou zde uvedeny podrobné popisy nástrojů použitých v této práci. Na konci teoretické části je sepsána kapitola, ve které se nachází stručný popis problematiky strojového učení, jsou zde podrobně popsány metody neuronových sítí, náhodného lesu a CHAID. Je zde také uveden matematický popis kritérií pro predikci dat.

V praktické části práce jsou prvně zkoumány možnosti výpočtů predikce na cloudových platformách Microsoft Azure a IBM Cloud. Je provedeno praktické odzkoušení všech nástrojů, které tyto cloudové platformy v rámci strojového učení nabízejí. Následně jsou zkoumány možnosti, jak lze vytvořené predikční modely v rámci cloud computingu nasadit. Z poznatků, nasbíraných při práci s cloudy, je sepsána kapitola, která kromě popisu jednotlivých nástrojů obsahuje i zkušenosti s užíváním cloudových platforem v delším časovém horizontu.

S desktopovou verzí nástroje SPSS Modeler jsou provedeny experimenty za účelem nalezení vhodného nastavení parametrů použitých predikčních metod. Dalším účelem experimentů je stanovení vhodné predikční metody pro finální nasazení při dlouhodobém výpočtu predikce v reálném čase.

V jazyce Python je sestaven kód, ve kterém jsou implementovány metody strojového učení náhodný les. Ke konektivě dat mezi PI Serverem a osobním počítačem je použita PI Web API. Též je v rámci streamování dat PI System použit nástroj PI Integrator for Business Analytics. V návaznosti na tento nástroj je zde popsáno použití Apache Kafka jako sprostředkovatele streamování dat.

Na závěr je v nástroji PI Vision vytvořena vizualizace provozně technických funkcí dvou bytů, které se nachází na budově CPIT TL3. Byty jsou vybaveny SmartHome systémy. V rámci vytvoření vizualizace jsou ve stejnojmenné kapitole popsány způsoby, jak vytvořit databázi PI Asset Framework a jak provádět konverzi datových typů PI Points. Na závěr je uveden popis vytvořené vizualizace. Vizualizace obsahuje i predikovaná data koncentrace oxidu uhličitého a kritéria této predikce.

2 Rešerše současného stavu vizualizace provozně technických funkcí v Inteligentní budově

C.M. Lin a M. T. Chen ve svém článku popisují implementaci inteligentního systému pro úsporu elektrické energie v domácnosti kombinováním inteligentních měřičů, inteligentních konektorů, inteligentních mobilních zařízení a databázového serveru. Zařízení využívají pro komunikaci moduly Zigbee. Uživatel může pomocí chytrého telefonu kontrolovat a ovládat provoz zařízení. Informace o spotřebě energie lze vzdáleně sledovat připojením inteligentního konektoru k internetu prostřednictvím Wi-Fi [1].

Weixian Li a jeho tým ve svém článku popisují vyvinutý systém detekce energie, zvaný inteligentní systém krádeže energie (SETS), který je založený na strojovém učení a statistických modelech. Tento systém je rozdělen do tří fází: predikční model (systém integruje různé modely strojového učení do jediného predikčního systému), primární rozhodovací model (založený na jednoduchém klouzavém průměru) a sekundární rozhodovací model. Výsledky simulace ukazují, že navrhovaný systém dokáže úspěšně detekovat s přesností 99,96 % [2].

Umar Ahsan a Abdul Bais ve své práci popisují návrh inteligentní distribuované architektury, zahrnující domácí senzory, které komunikují přímo s inteligentní bránou, nainstalovanou v domácnosti. Brána pak rozhodne, která data by měla být předána do centrálního procesoru pro další analýzu. Výpočty, jako je predikce využití energie pro domácí spotřebiče, lze efektivně provádět lokálně, zatímco centrální procesor lze použít pro koordinaci mezi různými místními procesory [3].

Ghassem Mokhtari a jeho tým ve svém článku popisují novou vrstvenou architekturu, určenou pro zpracování a správu velkého objemu dat inteligentních domácností budoucích let. Navrhovaná architektura je založená na Representational State Transfer (REST), zahrnuje sedm vrstev: fyzickou, výpočetní, síťovou, cloud computing, servisní, relační a aplikační. Navrhovaná architektura inteligentních domů může poskytnout všudypřítomné a sdílené datové prostředí [4].

Prabal Verma a Sandeep K. Sood se ve svém článku zabývají využitím IoT zařízení ve zdravotnictví. Říkají, že zpoždění, které je způsobené přenosem dat do cloudu a zpět do aplikace, je nepřijatelné. Navrhují vzdálené sledování zdraví pacientů v inteligentních domácnostech pomocí konceptu mlhového výpočtu na inteligentní bráně. Tento systém byl testován a jeho výsledky ukazují, že je schopen v reálném čase reagovat na zdravotní stav uživatele [5].

Lun-De Liao se svým týmem představují ve svém článku multifunkční, levný a flexibilní systém pro inteligentní ovládání domácnosti a monitorování životního prostředí. Navržený systém lze ovládat přes internet prostřednictvím mobilní aplikace pro Android. Systém přenáší data ze senzorů do cloudové platformy a může přijímat příkazy ze serveru, což umožňuje automatickou kontrolu mnoha zařízení [6].

Ying-Tsung Lee a jeho tým se ve svém článku zabývají ochranou dat a soukromím v IoT. Nastíňují řadu aplikací pro data získaná z inteligentní domácnosti. Aplikace využívají vyvinutého mechanismu ochrany soukromí. Mechanismus monitorování a ochrany kombinuje technologie zvyšující ochranu soukromí se strategií zachování soukromí. To vše od počáteční fáze návrhu systému, až po úplnou správu životního cyklu dat [7].

Jinsoo Han se svým týmem ve své práci pojednávají o nasazení obnovitelných zdrojů energie v chytrých domácnostech. Dle nich by systém pro správu domácí energie (HEMS) měl zvážit spotřebu i výrobu energie současně, aby se minimalizovaly náklady na energii. Dále navrhuje inteligentní architekturu HEMS, která zohledňuje spotřebu energie i generování současně. Tento systém využívá modulů ZigBee pro sledování spotřeby energie, pro monitorování vyrobené energie slouží PLC. Data se ukládají na domácí server, kde dochází k analyzování a rozhodování o spotřebě [8].

Pan Zhou a Zhong Guohui se ve své práci zabývají energeticky úspornými (ES) systémy založenými na IoT, které využívají automatizované porozumění lidskému chování a rozpoznávání činností. Tyto dosavadní systémy však trpí řadou problémů. Tudíž navrhuje online systém ES (PRCOES) na ochranu soukromí a rezidenční kontext v inteligentních domácnostech s podporou IoT. Prostřednictvím algoritmu učení je možné předpovídat energetickou náročnost, a tudíž docílit snižování spotřeby energie [9].

Abdulsalam Yassine, Singh Shailendra a Alarmi Atif ve svém článku navrhuje model, který využívá data z chytrých domácností, jako prostředek k učení a nacházení vzorců lidské činnosti pro aplikace ve zdravotnictví. Ten analyzuje časové vzorce spotřeby energie na úrovni spotřebičů, které přímo souvisí s lidskými činnostmi. Díky tomu je systém schopný rozeznat, zda si uživatel v daný den vařil, zda se sprchoval nebo zda nedošlo k zásadní změně v jeho denní rutině [10].

Basheer Qolomany se svým týmem se ve své práci zaměřují na oblast inteligentních budov se zvláštním zaměřením na roli technik strojového učení a analýzy velkých dat. Tento průzkum hodnotí současné trendy a výzvy, kterým čelí vývoj inteligentních budov [11].

Deamin Shin se svým týmem ve svém článku zabývají ochranou dat chytrých domácností v nastávající éře sítě 5G. Navrhuje bezpečnostní protokol, který se skládá z fáze inicializace a předání optimální trasy. Je navržen tak, aby poskytoval co nejvyšší ochranu soukromí. Bezpečnost je důkladně ověřena pomocí dvou formálních nástrojů pro analýzu zabezpečení: BAN-logic a AVISPA. Srovnávací analýza ukazuje, že navrhovaný protokol je lepší než jiné standardní protokoly [12].

A.R. Al-Ali a Imran A. Zuolkarnam se ve své práci zabývají úsporou energií na úrovni inteligentních měst. V tomto systému je každé domácí zařízení propojeno s modulem pro sběr dat, IoT zařízením. Naměřená data jsou přenášena na centralizovaný server pro další zpracování a analýzu. Tento systém využívá softwarové nástroje Business Intelligence (BI) a Big Data analytics, které zajišťují řízení spotřeby energie a uspokojení poptávky spotřebitelů. [13].

Ane Alberdi, Alyssa Weakley a Maureen Schmitter-Edgecombe se ve své práci zabývají využitím dat poskytovaných chytrou domácností k odhalení ranných příznaků Alzheimerovy choroby. Při vytváření analýzy byla po dobu dvou let shromažďována data z chytré domácnosti, přičemž každého půl roku byly prováděny zdravotní testy subjektů. S využitím těchto dat byly vytvořeny regresní modely pro predikci příznaků choroby měřených testy. Výsledky dvouletého testování ukazují, že všechny příznaky, jako jsou ztráta mobility, poznávání a deprese, lze předpovídat z dat chytré domácnosti [14].

Milica Lekić a Gordana Gardašević ve své publikaci představují implementaci aplikace Internet of Things (IoT), která zajišťuje měření teploty a vlhkosti prostřednictvím snímače DHT11, připojeného k Raspberry Pi. Data jsou následně přenášena do Cloud IIMB Bluemix. Implementace se provádí pomocí

programovacího systému Raspbian Stretch Lite na platformě Raspberry Pi a IBM Internet of Things založené na nástroji Node-RED, který je nainstalován na Raspberry Pi a IBM Cloud [15].

J. Koo a Y. G. Kim se ve své práci zabývají problematikou identifikací IoT zařízení (id), jelikož IoT platformy různých organizací (IBM Watson IoT, GS1, OneM2M, atd.) používají různé způsoby identifikací zařízení. Proto autoři analyzují existující způsoby, jak jsou zařízení identifikována a navrhnou konceptní model překladače ID IoT zařízení, který nazvali IoT DNS (Device Name System). Tento nástroj překládá různé typy ID zařízení do OneM2M ID formátu [16].

S. Marksteiner a spol. se ve své práci zabývají bezpečnostním rizikem komunikačních protokolů používaných v IoT. Představují přehled IoT aplikačních domén a diskutují o nejdůležitějších protokolech, používaných v komunikaci bezdrátových IoT zařízeních, kterými jsou KNX-RF, EnOcean, Zigbee, Z-Wave a Thread [16].

Marcin Bajer se ve své práci zabývá otázkami ohledně úspory energií, zvýšení pohodlí života a potenciálních hrozeb v inteligentních budovách. Autor se snaží odpovědět na otázky ohledně budoucnosti a dalšího směru inteligentních budov. Stručně se věnuje současným trendům v automatizačních systémech budov. Dále je zde popsána aktuální implementovaná řešení, zejména z portfolia ABB a potenciální rušivé technologie, které budou utvářet budoucnost inteligentních budov [18].

D. Bastos, M. Shackleton a F. El-Moussa se ve své práci zabývají otázkou bezpečnosti IoT. Předpokládají, že při nástupu Amazon Echo a zařízení Google, jako je Chromecast a Home, se počet zařízení připojených k IoT exponenciálně navýší. Upozorňují na hrozbu levných IoT zařízení, která mají téměř nulové zabezpečení. Tento dokument představuje komplexní přehled současných technologií IoT a bezpečnostních otázek se zaměřením na prostředí inteligentních domů a měst. Pojednává o možných řešeních pro zlepšení bezpečnosti internetu věcí [19].

Shubhradeep Nandi ve své práci popisuje využití nástroje IBM Watson v zabezpečovacích systémech. Tento systém by měl být nástupcem dnešních zabezpečovacích systémů, založených na biometrických datech, či nákladných bezpečnostních agentur. Jednalo by se o nákladově efektivní kognitivní předpokladový bezpečnostní systém [20].

S. Tanwar a spol. ve své práci popisují problematiku bezpečnostních systémů ve vztahu k IoT, inteligentním zařízením a internetu. Poukazují na různé problémy těchto zabezpečovacích systémů. V tomto dokumentu je popsán návrh pokročilého bezpečnostního výstražného systému založeného na internetu, který detekuje narušitele nebo jakoukoli neobvyklou událost. Jedná se o nízkonákladový bezpečnostní systém využívající PIR modul připojený na Raspberry Pi. Tento systém má minimální zpoždění při odesílání zpráv uživateli [21].

Sylvia Ilieva, Andrey Penchev a Dessislava Petrova-Antonova ve svém příspěvku věnuje návrhu rámce IoT pro integraci zařízení krátkého dosahu, tzv. SmartFW. Rámec smartFW funguje jako prostředník mezi integračními platformami používanými koncovými uživateli k řízení jejich inteligentních domů a připojených zařízení v těchto domovech. Má flexibilní architekturu, která je založena na standardu OSGi. V současné době podporuje komunikační protokoly zařízení Zigbee, EnOcean, KNX, X10, Z-Wave a aplikační protokoly Coap, MQTT a XAMPP [22].

Shivangi Vashi a jeho tým se ve svém článku zabývají bezpečností a architekturou IoT sítí. Jelikož je IoT technologií, která se rozvíjí po celém světě, je její bezpečnost velice důležitou záležitostí. V první části článku je poskytnut přehled architektury IoT pomocí Smart World. Druhá část se zabývá bezpečnostními výzvami v IoT, po nichž následují bezpečnostní opatření v IoT. Výzvy popsane v tomto dokumentu by mohly být předmětem pro budoucí práci v oblasti bezpečnosti internetu věcí [23].

A. A. N. Shirehjini se svým týmem ve své publikaci popisují uživatelské rozhraní, pomocí kterého je uživatel schopný manipulovat se zařízeními v inteligentní budově, aniž by je musel ovládat přes jejich IP adresy a přístupová rozhraní. Toto UI je založeno na 3D vizualizaci a přímém fyzickém spojení se zařízeními [24].

Ge Mengmeng, Hong, Jin B., Guttman Walter a Kim Dong Seong ve své práci popisují rámec pro modelování a posouzení bezpečnosti Internet of Things (IoT). Prostřednictvím toho rámce je možné hledání potenciálních scénářů útoků v rámci IoT a posouzení možných obraných strategií. Zde popisovaný rámec se skládá ze zpracování dat, generování bezpečnostního modelu, vizualizace zabezpečení, bezpečnostní analýzy a aktualizace modelu. Hodnocení rámce je v práci popsáno prostřednictvím scénářů napadení inteligentní domácnosti a monitorování systémů zdravotní péče [25].

Kumar K. Mathan a Venkatesan R. S. ve své práci popisují spolehlivý zdravotní monitorovací systém určený pro nízkonákladové monitorování pacientů. Projekt se zabývá bezdrátovými technologiemi, které je možné využít pro monitorování pacientů. Vizualizační stránkou je zde zobrazování zaznamenaných biomedicínských signálů na mobilních zařízeních s OS Android. Tato vizualizace je určena jak pro lékaře, tak pacienty, kteří mohou tato data sledovat v reálném čase. Tento koncept je představen jako cenově dostupný pro globální řešení zdravotní péče se schopností zpracování velkého množství biomedicínských signálů [26].

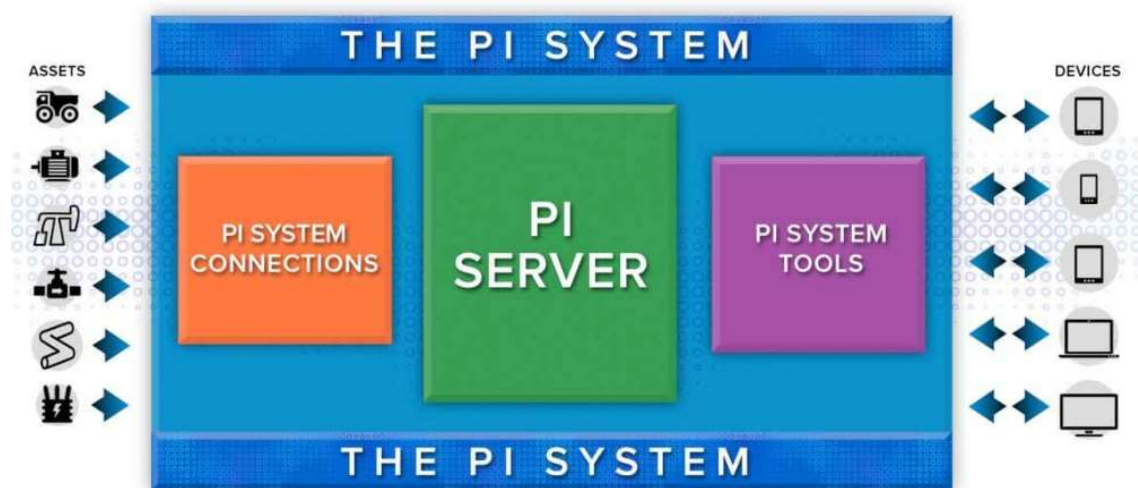
3 PI System

V této diplomové práci je použit PI system firmy OSIsoft, LLC. OSIsoft je americká firma, zabývající se od roku 1980 vývojem software PI System. Dnes je již součástí nadnárodní společnosti AVEVA. Pod souhrnným názvem PI system se nachází celá řada jednotlivých softwarů. Některé z nich budou v této kapitole podrobně popsány.

PI system je prostředkem pro správu dat a událostí v reálném čase, je nasazován v infrastruktuře podniků či inteligentních budov. Tento komplexní systém umožňuje snadné vyhledávání, ukládání, distribuci, analýzu a vizualizaci dat z různých datových zdrojů (kontrolní systémy, manuální vstupy, sensory, výpočty, SCADA systémy, ...). [27]

Základní prvkem PI System jsou zdroje dat, které máme v úmyslu sledovat a zaznamenávat jejich činnost. Jako zdroj dat si můžeme představit prakticky cokoli od textového souboru až po databázi. PI může být rozdělen na tři základní části:

- datová část (PI Interface),
- serverová část (PI Serve),
- klientská část (PI Clients).



Obr. 1 Architektura PI System [28]

3.1 Datová část

Reprezentuje fyzické propojení datových zdrojů s kontrolními systémy (SCADA, PLC, DDC, sensory, kontroléry atd). Může jít o propojení přímo s PI Serverem v případě manuálních vstupů, databázových systémů nebo historických datových uložišť. Tato oblast komunikuje s datovými zdroji. Pro zabezpečení datových zdrojů nabízí PI System několik úrovní zabezpečení. Mezi těmito datovými zdroji a PI Serverem probíhá datová komunikace. Ne však přímo, komunikují skrze PI Interface. Dále probíhá sběr a ukládání dat [28].

Připojení datových zdrojů:

- **Interfaces** – Jedná se o softwarové aplikace mezi datovými zdroji a PI Serverem. Tyto aplikace běží na operačních systémech Windows nebo ve speciálních případech na UNIXU. Systém, ve kterém jsou rozhraní implementována, se nazývá uzel rozhraní neboli interface node. Data jsou posílána z datových zdrojů přes Interface na PI Server. Většina rozhraní dokáže posílat data zpět do datových zdrojů. Získaná data z datových zdrojů se za pomoci Interfejsů ukládají v části PI Serveru zvané PI Data Archive. Zde mají data širokou škálu možností nastavení. Proto je samotná konfigurace náročnější. Každé zařízení má vlastní rozhraní související s rozdílným formátem dat. PI podporuje přes 400 různých rozhraní [28].
- **Connectors** – Konektory jsou podobné jako rozhraní. Kvůli menší možnosti nastavení jsou jednodušší na instalaci a správu. Zároveň díky automatickému modelování dat v PI Asset Frameworks zvyšují výkon systému [28].

3.2 PI Server

Jedná se o software pracující s daty, které jsou upravené způsobem popsaným níže. Data přicházejí na server v podobě událostí – PI Events. Ty jsou složeny z časové známky, hodnoty nebo status a dalších atributů. PI Server ukládá data z PI Interfaces a Connectors do PI data Archive, odkud jsou na vyžádání poskytována klientům. Způsob a forma uložení může být pro běžného uživatele nepohodlná a časově náročná, zvláště při objemných databázích. Proto byla přidána k PI Data Archive rozšiřující databáze PI Asset Frameworks [29].

PI Data Archive

Tato část sbírá, ukládá a distribuuje data v reálném čase a komunikuje s rozhraními. Je základním kamenem celého PI System. Právě tato část zpracovává data časového řádu, tzv. time-series data. Vlastností dat časového řádu je, že jejich objem velmi rychle roste. Právě rostoucí objem těchto dat způsoboval v minulosti komplikace a kladl velké nároky na jednotlivé databázové systémy (prostor na disku, rychlost systému atd.) [29].

Data jsou automaticky označena jedinečným atributem zvaným PI Point (dále jen Point). Ten může být typu Int32, Int64, Float32, Float64, String, Digital nebo BLOB. Dále obsahuje časovou značku a dalšími atributy definující jeho jednotlivé vlastnosti (datový zdroj, datový typ atd.). Uživatel musí zvolit jedinečný název atributu *Tag*, který je typu char. Takto uložená a komprimovaná data jsou jednoduše a rychle zpracovatelná pro PI [29].

PI Asset Framework (PI AF)

Jedná se o robustní nástroj, umožňující efektivní hierarchickou organizaci dat, modelování dat na základě assetů, integraci, upřesnění, distribuci, bližší analýzu dat a jejich zasazení do kontextu. Asset reprezentuje fyzický objekt, jenž je částí nějakého zařízení (senzor vlhkosti v místnosti, otáčkoměr ventilátoru atd.) [29].

Kompletní hierarchie v PI AF se sestává ze skupin elementů se specifickými vztahy, tzv. referencemi (rodič-potomek, kompozice, případně si uživatel může nadefinovat vlastní podle potřeby). Elementy obsahují množství atributů (název zařízení, přítok, teplota, hustota, matematický výpočet nebo PI

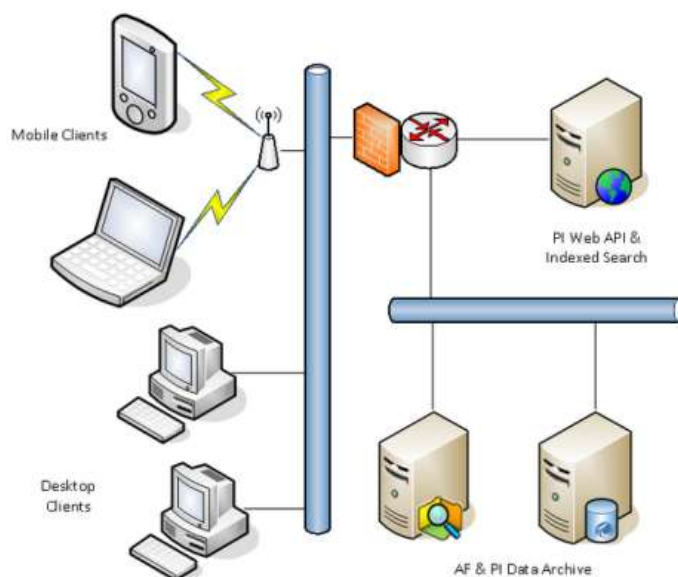
Point). K jednotlivým atributům můžeme přiřadit specifické metrické soustavy, které PI AF automaticky konvertuje podle potřeby [29].

3.3 PI Web API

Jedná se o RESTful webové rozhraní umožňující přístup do PI Sytem. Prostřednictvím PI Web API mohou klientské aplikace skrze webové požadavky provádět čtení i zápis time-series dat z PI Asset Framework a PI Data Archive. Rozhraní RESTful umožňují vývoj webových aplikací napříč platformami v mnoha různých programovacích jazycích. Jedná se o běžně používaný prostředek komunikace mezi klientem a serverem [30].

PI Web API verze 2019 nabízí následující způsoby autentifikace:

- BASIC,
- KERBEROS,
- BEARER,
- ANONYMOUS.



Obr. 2 Architektura nasazení PI Web API

Samotné rozhraní PI Web API používá pro přístup k datům sadu AF SDK, takže její funkčnost je podmnožinou AF SDK. To také znamená, že PI Web API má omezenější výkon a pole působnosti než AF SDK. Pro mnoho aplikací je však rozdíl ve výkonu nepatrný. PI Web API je mnohem flexibilnější, pokud jde o použité platformy a jazyky. AF SDK je knihovna .NET, takže vytvořená aplikace musí běžet ve Windows a v jazyce, který podporuje načítání .NET dll. Naopak PI Web API může používat jakoukoli platforma a jazyk, umožňující tvorbu HTTP(S) požadavků. [30]

Parametry URL

Parametry URL se používají k předání dalších hodnot metodě k řízení toho, co se vrací. Každá metoda přijímá konkrétní skupinu parametrů, z nichž některé mohou být vyžadovány a ostatní mohou být volitelné (mají defaultní hodnotu). Výběr důležitých URL parametrů [30]:

- WebId – Jedinečný identifikátor primárního zdroje.
- SelectedFields – Určuje, jaká pole mají být z odpovědi vynechána.
- MaxCount – Kontroluje maximální počet výsledků, které mají být navraceny (defaultně 1000).
- TimeZone – Jaké časové pásmo bude PI Web API interpretovat dodaná časová razítka.
- BufferOption – Řídí, zda při zápisu dat použít PI vyrovnávací subsystém.

WebID

Každý prostředek v rámci PI Web API má jedinečný identifikátor zvaný WebId. Tento identifikátor má každý prvek, ať se jedná přímo o Point nebo element databáze. Tyto identifikátory jsou trvalé a v průběhu času se nemění. PI Web API verze 2017 R2 představila novou verzi WebId 2.0, jenž dodržuje otevřený standard a umožňuje klientovi zkonstruovat WebIds. Při práci a vytváření požadavků PI Web API je WebID jedním z klíčových parametrů. [30] Ukázka WebID:

```
"WebId": "F1DSN1RwjYpuDkySLeSTwLU8YgUE1BRjE"
```

3.4 PI Integrator for Business Analytics

PI Integrator transformuje data systému PI do formátu, kompatibilního s nástroji Business Intelligence (BI). Nástroje BI nabízejí možnost provádět retrospektivní analýzy na mnohem větší sadě dat systému PI v reálném čase. Mezi nástroje BI patří cloudové nástroje jako je Microsoft Power BI, IBM Cognos Analytics, Amazon QuickSight a mnoho dalších [31].

Tento nástroj dále nabízí podporu pro streamovací platformy jako je Apache Kafka. Publikování dat na streamovací platformu umožňuje systému PI přenášet předdefinované balíčky dat časových řad spotřebitelům (konzumentům). Streamovací platformy pomáhají s provozováním a vytvářením modelů strojového učení [31].

PI Integrator for Business Analytics 2020 podporuje následující cíle:

- Amazon Kinesis, Amazon Redshift, Amazon S3;
- Apache Kafka, Apache Hive;
- Microsoft Azure (Data Lake storage, Event Hub, IoT Hub, SQL Data Warehouse, SQL Database);
- Google BigQuery, Google Cloud Pub/Sub, Google Cloud Storage;
- Microsoft SQL Server;
- Oracle database;
- SAP HANA ODBC;

Nativní data časových řad, kontext aktiv a kontext událostí jsou zpřístupněny prostřednictvím webové konfigurovaných pohledů. Data jsou dokonale dimenzionálně modelována, očištěna a prezentována s příslušnými metadaty, aby nástroje BI mohly správně procházet, dotazovat se a bez problémů využívat data systému PI. Data lze také přímo integrovat a načíst na platformy datových skladů. PI Integrator for Business Analytics eliminuje potřebu programování nebo znalostí jazyka SQL a spravuje celý životní cyklus dat [31].

PI Integrator for Business Analytics vyžaduje PI Asset Framework (PI AF) pro výběr dat systému PI k produkci dat. Data lze vyčistit pomocí různých filtrů, přidat k nim aktiva, události a časový kontext ze systému PI. Výsledná data jsou okamžitě použitelná v nástrojích BI bez dalších úprav [31].

3.5 PI Vision

Je webový nástroj pro rychlý a snadný přístup k datům PI. Prostředí tohoto nástroje je jednoduché a intuitivní. Vytváření displejů je založeno na práci drag-a-drop. Vytvořené displeje je možné ukládat a sdílet s ostatními uživateli. Do prostředí PI Vision se dají importovat displeje vytvořené v PI ProcessBook. Při vytváření displejů jsou k dispozici čtyři základní soubory nástrojů [32]:

- assets,
- výpočty (Calculations),
- grafické prostředky (Graphic library),
- události (Events).

Assets

V této záložce se nachází přístup ke všem assetům, které je možné v rámci PI Vision používat. Data jsou zde rozdělena do struktur jednotlivých databází (ty lze měnit a vytvářet v nástroji PI System Explorer). Jednotlivá data mohou být vizualizována v různých podobách:

- trend,
- hodnota,
- tabulka,
- horizontální, vertikální, kruhový ukazatel,
- XY graf,
- porovnávací tabulka.

Výpočty

Nově je v PI Vision možné provádět jednoduché kalkulace, a to v reálném čase. Dostupnými výpočty jsou: vyhledání minima a maxima (Min/Max), výpočet aritmetického průměru (Avg), výpočty zahrnující základní matematické operátory, jako jsou +, -, * a /. Výpočet lze aplikovat přímo na tagy umístěná na displeji a výsledky okamžitě vizualizovat.

Grafické knihovny

PI Vision nabízí širokou paletu již vytvořených grafických prvků. Ty lze jednoduše využít při vytváření displejů. Jsou zde prvky od obyčejných šipek a tlačítek, až po grafické zpracování částí budov, celých budov, vozidel, prvků těžkého průmyslu nebo například vzduchotechnických zařízení. Vybraný prvek stačí přetáhnout do prostoru vytvářeného displeje. Díky tomu lze velmi rychle a efektivně vytvářet nejrozumnější grafická prostředí pro vizualizaci dat.

Popis prostředí vytváření displejů

1. Galerie symbolů (Symbol gallery)
2. Výpočty (Calculations)

3. Knihovny grafických prvků (Graphics Library)
4. Události (Events)
5. Seznam assetů (Assets pane)
6. Seznam atributů (Attributes pane)
7. Časové rozlišení (Time bar control)
8. Srovnání obrazovky na velikost displeje (Fit all and zoom)
9. Uložit (Save button)
10. Přechod do návrhářského módu (Design mode button)
11. Seznam použitých assetů (Asset list)



Obr. 3 Popis rozmístění prvků pro vytváření displeje v PI Vision [32]

3.6 PI ProcessBook

Umožňuje uživatelům okamžitý přístup k datům jejich vizualizaci v reálném čase. Taktéž je možná vizualizace a prohlížení si historických dat. V PI ProcessBook se využívají integrace PI Asset Frameworks, tudíž je možné vytvořit jeden model pro různá zařízení, díky čemuž mohou být data lehko aktualizována a dynamicky analyzována [34].

Nespornou výhodou tohoto nástroje je vektorová grafika, díky které je možné vytvořit vizuálně atraktivní prostředí. Vytvořené displeje je možné dle potřeby uživatelů kdykoli upravit. Data na displejích jsou zobrazována v reálném čase. To znamená, že jsou k dispozici stále aktuální data, se kterými je možné pracovat.

Jakákoli úprava displejů je velmi jednoduchá a intuitivní. Pomocí menu a panelů nástrojů je úprava možná i bez podpory odborníků. Vytvořené displeje je též možné sdílet s dalšími uživateli aplikace nebo jejich uložení a znovu načtení v případě potřeby.

Aplikace upravuje časová pásma podle polohy daného uživatele, tudíž každý uživatel má k dispozici data s daty, odpovídající jeho časovému pásmu.

Důležitou součástí PI ProcessBook jsou alarmy. Jedná se o kritické hodnoty, na které systém upozorní v případě jejich překročení. Historie těchto alarmů se ukládá na server, takže je možné alarmy zpětně procházet [34].

3.7 Způsoby uložení dat

Pro efektivní uložení a doručení data klientovi na jeho vyžádání, používá PI při organizaci dat tyto části: PI AF Asset, PI Asset Attribute a PI Point (PI Tag) [33].

PI Asset Framework

Reprezentuje jednotlivé datové zdroje v podobě stromové struktury. Tyto assety nebo též elementy můžou představovat např. celý objekt rozdělený na části (na poschodí a ta jednotlivé místnosti a chodby) [33].

PI Asset Attributes

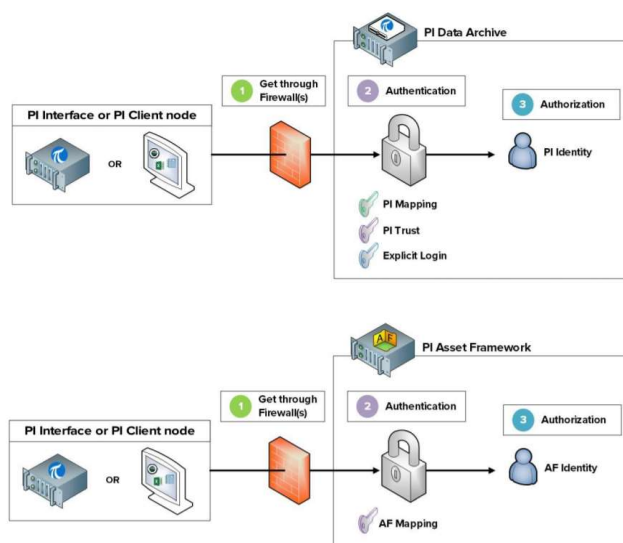
Popisují důležité vlastnosti jednotlivých elementů. Například tlak, teplota, rozměry místnosti nebo nádrže, počet otáček za minutu nebo datum poslední kontroly. Jedná se o různé typy hodnot, např.: konstanty, výsledky výpočtů, externí databázové zdroje a PI Points (PI Tags) [33].

PI Point

Též zvaný PI Tag. Je základní jednotkou PI, pomocí které se ukládají data v reálném čase. Reprezentuje jakákoli datový tok. Může se jednat o hodnotu datového zdroje v čase (přítok, objem nádrže, teplota atd.), případně o výsledek nějakého výpočtu (minimum, průměr atd.). Odkazem na tyto Pointy je název PI Tag. Tagy představují data v různých datových typech (Digit, Float, Integer, String, Timestep nebo Blob). Tagy mají atributy, které je podrobně popisují a odlišují od ostatních, například: unikátní název, ID, popis, Timestamp a Point source, který odkazují na zdroj dat [33].

3.8 PI System Security

V PI Serveru mohou být uložena citlivá a tajná data, tudíž je důležité, aby bylo vše správně zabezpečeno a ani aby se uživatelé PI System nedostali do míst, kam nemají oprávnění. Z těchto důvodů PI Server nabízí celkem pět různých konfiguračních částí, které zajišťují bezpečnost a uložení dat.



Obr. 4 Znáznornění zabezpečení připojení do PI Data Archive a PI Asset Framework [37]

PI Firewall

Je první vrstvu zabezpečení, poskytuje přístup založený na IP adrese nebo názvu PC. Záznamy databáze PI Firewall jsou uloženy do dvojic. První člen představuje adresu, druhý člen představuje bezpečnostní povolení. Ty určují, které IP adresy nebo IP adresy masky mají povolení pro připojení [37].

PI Identities

Identita PI AF představuje sadu přístupových oprávnění na serveru PI AF. Například identita může mít povoleno čtení Tagů, ale nemůže je vytvářet, mazat ani přepisovat. Další skupina (především admini) mají ke všem těmto úkonům oprávnění. Členům skupin Windows, které jsou mapovány na identitu PI AF, se automaticky udělují přístupová oprávnění pro danou identitu PI AF [36].

PI Trust

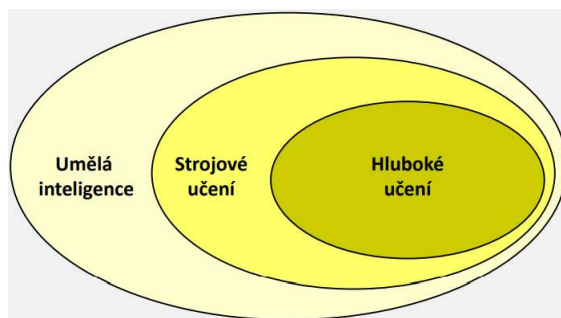
PI Trust se používá především pro zabezpečení připojení PC nebo aplikací než přímo s uživatelem. Většinou se jedná o aplikace, které jsou k PI System připojeny nepřetržitě (dny až týdny) a proto je tradiční způsob ověření identity nedostačující. PI Trust mapuje aplikaci, název hostitele, Windows doménu a/nebo uživatelské jméno s PI User. Princip činnosti PI Trust je jednoduchý. Aplikace najde PI Server. Když se PI aplikace připojí k PI serveru, odešle mu informace o bezpečnosti. Které informace bude aplikace posílat na server, závisí na použitém komunikačním protokolu. V případě PI Web API se posílá PI adresa připojeného počítače a první čtyři písmena názvu aplikace ukončené písmenem E [37].

PI Database Security

Jedná se o zabezpečení kontrolující, které PI Identities vytvořené v PI System můžou editovat (vytvářet a mazat) data. Defaultní nastavení je zde takové, že pouze administrátorský účet má oprávnění pro vytváření, editaci a mazání PI dat [37].

4 Strojové učení

Strojové učení je podoblastí umělé inteligence. Jedná se o problematiku použití matematických modelů, mající schopnost samo nastavení svých vnitřních parametrů, tak aby dosáhly požadovaného výstupu – mají schopnost učit se. Strojové učení pracuje s obrovským množstvím dat, jež analyzuje, uspořádá do souvislostí a následně vyhodnotí s cílem pochopení principů reálného světa. Samo s nimi nedokáže dále pracovat, o to se potom stará umělá inteligence. Strojové učení je tedy nástrojem umělé inteligence [38].



Obr. 5 Návaznost oblastí umělé inteligence [38]

Z matematiky či fyziky jistě každý zná výpočty na základě vzorců. Vzorec pevně definuje postup, jakým se výsledek počítá. Pro určení výsledku je nutné znát vzorec a všechny proměnné, výsledkem je vypočtená hodnota. Strojové učení funguje jinak. Do strojového učení vstupují data společně s výsledky (učení s učitelem) a výstup je algoritmus, který popisuje, jak výsledek predikovat. Výsledný algoritmus se ukládá do modelu, říkáme mu naučený model [38].

Základní rozdělení algoritmů učení:

- s učitelem,
- bez učitele,
- kombinované učení s učitelem i bez učitele,
- zpětnovazební učení.

Druhy úloh řešených strojovým učním:

- Klasifikace – Rozpoznávání objektů a roztřídování do tříd.
- Regrese – Analýza dat, hodnoty výstupu jsou určeny na základě vstupního vektoru hodnot.
- Shlukování – Data jsou shlukována a tříděna na základě podobnosti bez znalosti jejich obsahu.
- Asociace – Mezi daty jsou hledány souvislosti a jsou skládány do uceleného konceptu.

4.1 Hluboké učení

Hluboké učení neboli Deep Learning (DL), je specializovanou součástí strojového učení, využívající neuronové sítě. Neuronové sítě jsou podrobně popsány níže v této kapitole. Obecně se zabývá složitějšími a náročnějšími úlohami než strojové učení [38]. Hluboké učení je například využíváno automatickými překladači textu (Google Translate) nebo virtuálními asistenty Siri a Alexa.

4.2 Popis modelů strojového učení použitých v této práci

Zde jsou popsány modely strojového učení, které jsou použity v této práci. Dle výsledků experimentů bylo dosaženo nejpřesnější predikce s modely založenými na regresních stromech. V podkapitolách jsou zde popsány modely náhodného lesu – zástupce binárních rozhodovacích stromů a metoda CHAID – zástupce nebinárních rozhodovacích stromů. Neuronové sítě jsou zde popsány jako zástupci hlubokého učení.

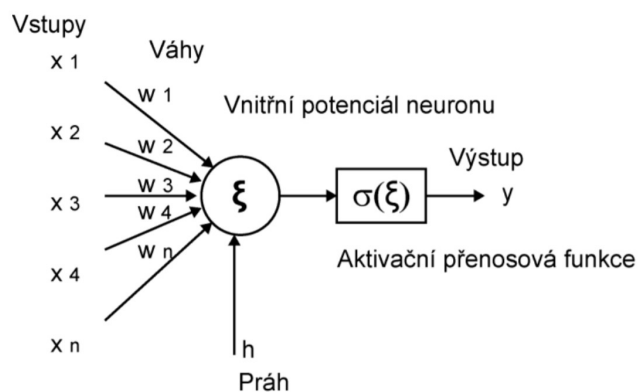
4.2.1 Neuronové sítě

Neuronové sítě jsou jedním z výpočetních modelů používaných v umělé inteligenci. Koncept neuronových sítí je inspirován biologickou předlohou – biologickou neuronovou sítí. Ta je složena z velkého množství neuronů, jejichž klíčovou vlastností je schopnost komunikace mezi sebou. Každá neuronová síť je mimo jiné charakterizována svým topologickým uspořádáním, typem neuronů nebo style adaptačního procesu při učení. Výsledná funkce neuronové sítě je dána způsobem propojení jednotlivých neuronů, vahami těchto propojení a způsobem činnosti jednotlivých neuronů (jejich aktivační funkcí) [41].

Základními typy jsou vícerozměrné dopředné neuronové sítě a neuronové sítě se zpětnou vazbou. V této práci jsou jako výpočetní model použity vícerozměrné dopředné neuronové sítě, tudíž se zde budou věnovat pouze jejich podrobnějšímu popisu.

Neuron

Neuron je složen z několika částí: Synapse ohodnocené vahami w (přivádějí vstupy x do těla neuronu), vlastní tělo neuronu, ve kterém je získáván vnitřní potenciál neuronu ξ . Dále je to blok aktivační přenosové funkce σ , a nakonec výstup neuronu y . Do těla neuronu kromě vstupních synapsí vstupuje konstantní hodnota prahu h . Schématický model neuronu je zobrazen na následujícím obrázku.



Obr. 6 Model neuronu [39]

Vstupem neuronu je vektor o n prvcích, které jsou obecně reálné, stejně tak i hodnota prahu. Avšak hodnota prahu je často modelována jako jeden ze vstupů. Váhy jednotlivých vstupů jsou opět reálná čísla, mohou být tedy kladné i záporné a modelovat tak aktivační i inhibiční synapse. Vnitřní potenciál neuronu lze při vložení vstupního vektoru x získat dle vzorce [39]:

$$\xi = \sum_{i=1}^n w_i x_i - h \quad (1)$$

Každý prvek vstupního vektoru je vynásoben příslušnou vahou, všechny tyto násobky jsou v těle neuronu sečteny, od tohoto součtu je odečtena hodnota prahu. Tato hodnota vnitřního potenciálu se následně stává argumentem obecně nelineární aktivační přenosové funkce.

Jak již bylo zmíněno výše, hodnota prahu se obvykle modeluje jako jeden z prvků vektoru vah. Vstupní vektor i vektor vah je rozšířen o nultou pozici, kdy hodnota vstupního vektoru na pozici nula je rovna 1 a nulová pozice vektoru vah je rovna záporné hodnotě prahu (-h). V takovém případě se práh stává jednou z vah a v průběhu trénování podléhá adaptaci.

Výstup neuronu je poté možno vyjádřit vztahem [39]:

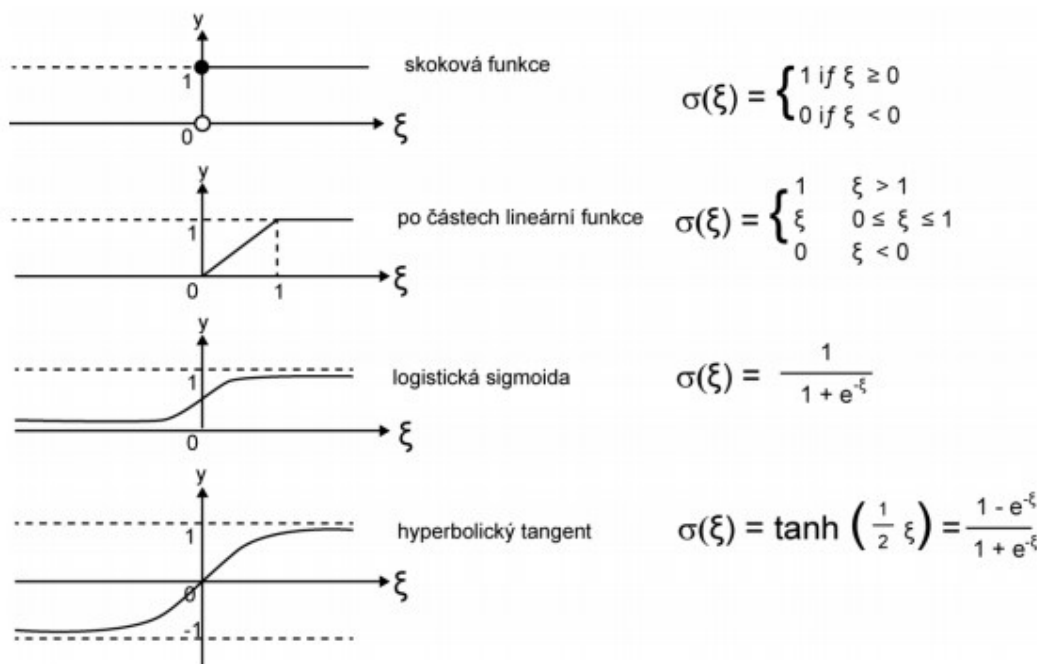
$$y = \sigma(\xi) = \sigma(\sum_{i=0}^n w_i x_i),$$

kde $x_0 = 1$ a $w_0 = -h$ (2)

Aktivační přenosová funkce σ je obecně nelineární funkcí transformující hodnotu vnitřního potenciálu neuronu. Pokud uvažujeme její nejjednodušší typ, ostrou nelinearitu, platí pro ni [39]:

$$\sigma(\xi) = \begin{cases} 1, & \xi \geq 0 \\ 0, & \xi < 0 \end{cases} \quad (3)$$

Aktivačních přenosových funkcí je celá škála. Ty nejčastěji používané v dopředných vícevrstvých neuronových sítích jsou zobrazeny na obrázku níže:



Obr. 7 Aktivační přenosové funkce [39]

Učení neuronových sítí

Proces učení je nazýván adaptační dynamikou neuronové sítě. Při tomto procesu jsou parametry neuronů nastavovány tak, aby výstupem neuronové sítě byl požadovaný výsledek. Parametry, které podléhají adaptaci, jsou především váhy vstupních synapsí neuronů (taktéž synapse představující práh). Samotné učení obstarává adaptační algoritmy. Ty při procesu učení postupují podobně jako člověk. Hledají tedy řešení na základě porovnávání výstupu se známými výsledky.

Učení s učitelem

Při učení s učitelem jsou do adaptačního algoritmu vloženy informace o požadovaném výstupu. Máme tedy množinu dvojic, které jsou tvořeny vstupy a k nim přiřazenými odpovídajícími korektními výstupy řešené úlohy. Ve většině případů je tato množina rozdělena na dvě části. Jedna část slouží pro trénování, druhá pro testování. Poměr rozdělení vstupní množiny není nijak pevně určen, volí se s ohledem na charakter řešené úlohy a na charakter dat množiny. Bývá však pravidlem, aby trénovací množina byla větší testovací, 70 % – 80 % [39].

Při učení jsou do neuronové sítě vpouštěny jednotlivé vstupy, je sledován jejich výstup, který je porovnán s požadovaným výstupem. Na základě odchylky mezi požadovaným a pozorovaným výstupem jsou upravovány hodnoty vah synapsí. Vstupní data jsou neuronové síti předkládána v intervalech, nazývajících se epochy. Epocha představuje projití celé trénovací množiny. Pro naučení sítě třeba desítky, v některých případech i tisíce epoch. Stav, kdy neuronovou síť považujeme za naučenou,

může být dán různými způsoby. Například dojde ke splnění nastavené minimální chyby transformace nebo je dosažen nastavený počet epoch [39].

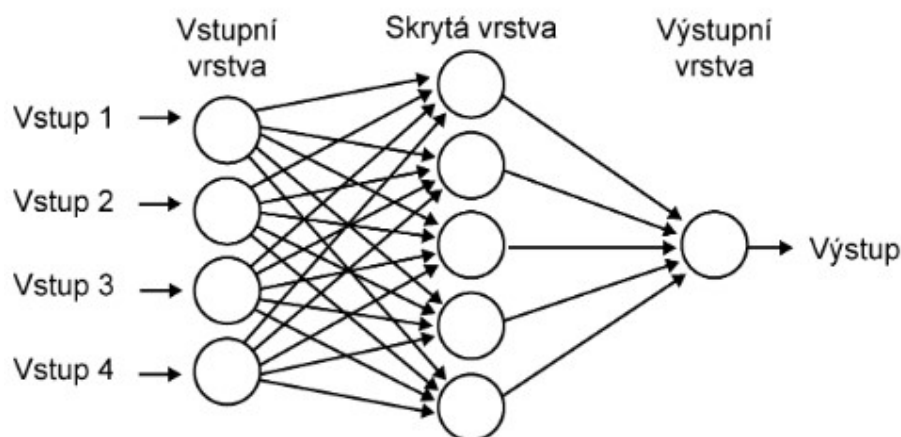
Výkonost naučené neuronové sítě se po dokončení adaptačního cyklu určuje pomocí testovací množiny. Výsledky testování naučené sítě jsou představovány v pohodě různých kritérií, jako je střední absolutní chyba, střední kvadratická chyba nebo lineární korelace. Všechna tato kritéria zohledňují odchylku od požadovaného výstupu [39].

Učení bez učitele

Oproti učení s učitelem, nemá učení bez učitele žádné kritérium správnosti hledané transformace vstupních dat, výsledek není znám. Tato metoda učení pracuje na principu shlukování, kdy jsou prvky vstupního vektoru dat shlukovány na základě jejich podobnosti. Na základě parametrů předkládaných vzorků vstupních dat se provádí jejich třídění do skupin, bez možnosti posouzení správnosti zatřídění. Počet hledaných skupin je možné předem zadat [39].

Dopředná neuronová síť

Vícevrstvá neuronová síť (Obr. 8) je tvořena jednou nebo více skrytými vrstvami neuronů umístěných mezi vstupní a výstupní vrstvou. Výstupy neuronů jedné vrstvy jsou přivedeny na vstupy všech neuronů následující vrstvy. Signál se šíří pouze jedním směrem – ze vstupní vrstvy na výstupní. Následující obrázek popisuje třívrstvou dopřednou síť. Vstupní vrstva slouží však pouze k opakování vstupních dat. Lze tedy tuto síť popsat též jako dvouvrstvou [40].



Obr. 8 Uspořádání neuronů do vrstev v dopředné neuronové síti [40]

Počet neuronů ve vstupní vrstvě je dán počtem skupin vstupní množiny, počtem vstupů matematického modelu. Počty neuronů ve skrytých vrstvách jsou určeny dle náročnosti a charakteru požadované úlohy. Standardně je síť učena za pomoci učení s učitelem. Jako aktivační funkce je obvykle volena funkce lineární nebo sigmoidální, proces učení je u těchto sítí poměrně dlouhý. Tyto sítě se především využívají při predikci, klasifikaci a aproximaci [40].

4.2.2 Náhodný les

Náhodný les je pokročilá implementace pytlovacího algoritmu se stromovým základním modelem. Jedná se o metodu kompletního učení pro účely klasifikace či regrese, které fungují na základě sestavování rozhodovacích stromů ve fázi tréninku [42].

- Klasifikační les je klasifikační model tvořený kombinací určitého počtu klasifikačních stromů. Každý strom přiřazuje hodnotě vektoru prediktorů nějakou třídu. Výsledná klasifikace je dána hlasováním nebo jako průměr pravděpodobnosti (zastoupení kategorie v terminálním uzlu).
- Regresní les Je tvořen několika regresními stromy. Výsledná regresní funkce je definována jako vážený průměr regresních funkcí několika stromů.

Regresní strom

Regresní stromy se dělí na dva základní skupiny – stromy CART (*Classification and Regression Tree*) a MARS (*Multivariate Addaptive Regression Splines*). V této práci jsou pro predikci použity metody založené na stromech CART, tudíž se zde budu věnovat především jejich popisu.

CART

Tato metoda byla vyvinuta, stejně jako mnoho jiných, Braimanem. Její využití je možné jak pro řešení regresních, tak i pro řešení klasifikačních úloh [43].

Algoritmus CART

Datový soubor $L = \{(x_1, y_1), \dots, (x_i, y_i)\}$, který obsahuje vektor vysvětlujících proměnných $x_i = (x_{i1}, \dots, x_{id})$ a vektor vysvětlované proměnné y [43].

Algoritmus se skládá ze čtyř kroků [43]:

- 1) Nalezení nejlepšího dělení pro každý prediktor – pro každou vysvětlující proměnnou X_j v každém z uzlů t se najde nejlepší dělení.
- 2) Výběr nejlepšího dělení daného uzlu t - mezi nejlepšími děleními pro jednotlivé vysvětlující proměnné, které jsme našli v bodě jedna, vybereme to, které nejvíce snižuje nepřesnost stromu
- 3) Dělení uzlu - pozorování z uzlu rozdělíme do dvou následníků dle dělení vybraného v předešlém kroku.
- 4) Přiřazení hodnoty - kroky 1-3 se opakují, dokud uzel není označen jako konečný, tj. z uzlu se stává list. Až se to stane, algoritmus přiřadí danému listu nějakou hodnotu. Algoritmus končí, když jsou všechny uzly označené jako listy.

Les

V regresním lese je vytvořeno L stromů T_1, \dots, T_L . Výsledek regrese je poté tvořena regresní funkcí $d_A(x)$, která vzniká vhodným zkombinováním regresních (popř. klasifikačních) funkcí $d_1(x), \dots, d_L(x)$ jednotlivých stromů. Jednoduchým a základním způsobem je u regresních lesů aritmetický průměr, u klasifikace je to většinové hlasování. Dáno vztahem [42]:

$$d_A(X) = C_{i^*}, \text{ pokud } \#\{j; d_j(x) = C_{i^*}\} = \max_{i=1, \dots, k} \#\{j; d_j(x) = C_i\} \quad (4)$$

Symbolem # je značen počet prvků. Složitějším způsobem je zohlednění váhy jednotlivých stromů. Hodnota váhy jim je přiřazována za základě jejich přesnosti.

Metody vytváření stromů

Bagging

Název této metody je tvořen akronymem „bootstrap aggregating“.

Z trénovacího datového souboru, který obsahuje data $X=y_1, \dots, x_n$ a odpovědi $Y = y_1, \dots, y_n$, jsou opakovaně (B krát) vybírány náhodné vzorky o velikosti n (bootstrapových výběrů). Každý z těchto vzorků je použit pro sestavení jednoho regresního (klasifikačního) stromu. Výsledný regresní les je potom dán aritmetickým průměrem dílčích regresních funkcí. Pro neviditelné vzorky x' lze vypočítat aritmetický průměr dle [42]:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad (5)$$

Pro klasifikační lesy je výsledek dán většinovým hlasováním.

Jednotlivé bootstrapové výběry nejsou nezávislé jako například u křížové validace, do bootstrapového výběru jsou některá pozorování vybrána opakovaně a některá naopak vůbec. Vzhledem k tomu, že počet opakování má pro jednotlivé pozorování z L asymptoticky (pro $n \rightarrow \infty$) Poissonovo rozdělení se střední hodnotou 1, je pravděpodobnost, že pozorování nebude vůbec vybráno, je přibližně $e^{-1} \approx 0,37$. Z tréninkového datového souboru je tedy využito 63 % dat [42].

Boosting a arcing

Metoda označovaná též jako *Boosting* či *AdaBoost*, pracuje s algoritmem pro vytváření stromů, který konstruuje posloupnost rozdílných modelů T_1, \dots, T_L s klasifikačními funkcemi $d_1(x), \dots, d_L(x)$ tak, že se podle předcházejících výsledků postupně upravují váhy případů. Nejprve je použit váhový vektor w_1 , který je zadán uživatelem, a je vytvořen model T_1 . V následujících krocích je ke konstrukci modelů T_i ($i=2, \dots, L$) použit váhový vektor w_i , který je získán úpravou vektoru w_{i-1} . Tato úprava je provedena zvýšením váhy v případě chybně klasifikovaného modelu T_{i-1} , snížením v případě správně klasifikovaných [42].

Random forest

Opět jsou zde trénovací soubory výběrem z datového souboru L . Při volbě větvení pro daný uzel se z m prediktorů X_1, \dots, X_m , které jsou k dispozici, nejdříve náhodně vybere některých m_0 . Následně se nejlepší větvení hledá tradičním způsobem, ale jen mezi těmi větveními, která jsou založena na vybraných m_0 veličinách. Pěstují se velké stromy, které se neprořezávají [42].

Náhodný výběr prediktorů výrazně urychluje výpočty. Také zvyšuje přesnost, především pro klasifikační úlohy. Pro regresní úlohy jsou ještě mnohem výhodnější modifikace této metody (BoostTrees) [42].

4.2.3 CHAID

Je jedním ze dvou (CHART a CHAID) základních metod vytváření rozhodovacího stromu, řadí se mezi klasifikační analýzy. CHAID je zkratka pro Chi-Squared Automatic Interaction Detection, neboli “automatizovaná chí-kvadrát detekce interakcí”.

Algoritmus analýzy CHAID se skládá ze tří částí:

- I. **Slučování** – První fáze algoritmu CHAID je založena na předpokladu, že kategoriální vysvětlovaná proměnná má d kategorií a sledovaný nominální nebo ordinární prediktor má c kategorií. Tato data se slučují do kontingenční tabulky, která má rozměr $c \times d$. Nejprve jsou sloučeny kategorie podobných prediktorů. Každý prediktor vysvětlované proměnné je spojen s kategoriemi, které vykazují nízkou statistickou odlišnost proti vysvětlované proměnné. Odlišnost kategorií prediktorů se zjišťuje na základě výpočtu chí-kvadrát testu o nezávislosti, přičemž využíváme hodnoty z kontingenční tabulky. Testuje se statisticky významná odlišnost hodnot vůči vysvětlené proměnné. U tohoto je postup pro ordinární a nominální proměnnou odlišný. U nominálního prediktoru jsou testovány všechny dvojice kategorií bez ohledu na pořadí. U ordinárního prediktoru se testují pouze dvojice po sobě jdoucích kategorií [44].

V případě, že chí-kvadrát test o nezávislosti hodnot v kontingenční tabulce neprokáže na zvolené hladině statisticky významnou odlišnost mezi kategoriemi, mohou být tyto kategorie sloučeny do jedné tzv. redukované kategorie. Toto slučování probíhá v každé iteraci, nejdříve jsou pro všechny možné dvojice kategorií vypočteny p-hodnoty v kontingenční tabulce o velikosti $2 \times d$ [44]:

$$p^{(i)} = P\left(\chi^2^{(i)} < G^{(i)}[d - 1]\right), \quad i = 1, 2, 3, \dots, s \quad (6)$$

Vypočítané p-hodnoty jsou mezi sebou porovnány, dvojice s nejvyšší p-hodnotou jsou sloučeny.

Z důvodu těchto slučování je nutné test chí-kvadrát o nezávislosti hodnot v kontingenční tabulce provádět při každé iteraci, jelikož vzniklá redukovaná kategorie nemá přiřazenou hodnotu chí-kvadrát testu a není známa její statistická odlišnost. Toto probíhá do chvíle, kdy jsou na zvolené hladině významnosti všechny kategorie prediktoru statisticky odlišné. V tu chvíli je vytvořen nový prediktor s redukovaným počtem kategorií $r \leq c$ [44].

- II. **Rozdělení** – Ve druhé fázi jsou kategorie prediktorů rozděleny do jednotlivých uzlů a je zahájeno vytvoření samotného stromu. V této fázi nám původní p-hodnota nestačí, jelikož nezohledňuje počet kategorií. Je tedy vypočítána nová p-hodnota pomocí tzv. Bonferroniho úpravy. Tato úprava spočívá přenásobením původní p-hodnoty z redukované kontingenční tabulky $r \times d$ Bonferroniho multiplikátorem. Tvar Bonferroniho multiplikátoru je u ordinárního a nominálního prediktoru odlišný [44].

Pro nominální prediktor má tvar [44]:

$$B_{nom} = \sum_{i=0}^{r-1} (-1)^i \frac{(r-i)^c}{i! (r-i)!} \quad (7)$$

Pro ordinární prediktor je tvar následující [44]:

$$B_{ord} = \binom{c-1}{r-1} \quad (8)$$

Na základě upravené p-hodnoty chí-kvadrát testu o nezávislosti v redukované kontingenční tabulce $r \times d$ je zvolen prediktor, který nejlépe diferencuje hodnoty vysvětlované proměnné. Dále dle prediktoru s nejnižší upravenou p-hodnotou se vytváření větvení, přičemž každá kategorie bude tvořit jeden uzel [44].

III. **Zastavení** – První a druhá fáze algoritmu se opakuje do doby, než jsou splněny určité podmínky a běh algoritmu je zastaven [44].

- Pro žádný z uzlů již neexistuje další prediktor (vysvětlující proměnná), jehož upravená p-hodnota by byla nižší než nastavená hladina významnosti.
- Byl dosažen maximální nastavený počet úrovní stromu.
- V žádném dalším novém uzlu již nelze splnit předem nastavenou podmínku pro minimální počet pozorování.
- Všechny záznamy v uzlu mají stejnou hodnotu vysvětlované proměnné.
- Všechny záznamy v uzlu mají stejnou hodnotu pro každý prediktor.

Některé podmínky ukončení běhu algoritmu je možné nastavit před zahájením analýzy. Tudíž je důležité před spuštěním samotné analýzy věnovat dostatečnou pozornost správnému nastavení těchto parametrů (hladina významnosti, maximální počet úrovní stromu, minimální počet pozorování).

SPSS Modeler nabízí dvě varianty tohoto modelování – CHAID a tzv. vyčerpávající CHAID. Přičemž vyčerpávající CHAID je modifikace CHAID, která provádí důkladnější práci při zkoumání všech možných rozdělení pro každý prediktor, ale výpočet trvá déle [45].

4.3 Kritéria přesnosti predikce

Přesnost naučeného modelu se hodnotí dle vypočtených kritérií. Kritérií je mnoho, zde jsou uvedeny ty, které jsou použity v této práci.

Legenda ke všem níže uvedeným vzorcům:

- y_i je i-tý prvek vstupního datového souboru
- \hat{y}_i je i-tý prvek predikovaného souboru hodnot
- n je počet prvků v datovém souboru

Střední absolutní chyba (MAE)

Střední absolutní chyba představuje aritmetický průměr absolutních hodnot rozdílů mezi naměřeným a predikovaným průběhem. Jeho hodnota je vyjádřena ve stejných jednotkách jako jednotky naměřené veličiny [46]. Počítá se podle vzorce (9):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

Střední kvadratická chyba (MSE)

Střední kvadratická odchylka neboli rozptyl je používán v pravděpodobnosti a statistice. Jedná se o rizikovou funkci, která odpovídá aritmetickému průměru druhých mocnin rozdílů mezi hodnotami naměřenými a predikovanými [47]. Je vyjádřena vzorcem (10):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (10)$$

Střední absolutní procentuální chyba (MAPE)

Střední absolutní procentuální chyba udává míru přesnosti predikce. Jendou z nevýhod tohoto kritéria je použitelnost pouze pro datové soubory, které neobsahují nulové hodnoty. Je třeba tuto výjimku při implementaci MAPE ošetřit. Čím je hodnota MAPE nižší, tím je predikce přesnější. Dána vzorcem [48] (11).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (11)$$

Hlavní střední kvadratická chyba (RMSE)

Někdy též označována jako kořenová střední odchylka, je druhá odmocnina čtvercové ztráty a má stejné jednotky jako popisec, poskytuje větší váhu větším rozdílům. Je vždycky nezáporné a hodnoty blíž až 0,00 jsou lepší [49]. Je vyjádřena vzorcem (12):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (12)$$

4.3.1 Linear Correlation (LC)

Lineární korelace popisuje vzájemný vztah mezi dvěma procesy nebo veličinami, v tomto případě dvěma spojitými signály. Odpovídá stupni závislosti (korelace) mezi dvěma proměnnými. Je vyjádřena vztahem [50]:

$$LC = \frac{n \sum_{i=1}^n y_i \hat{y}_i - \sum_{i=1}^n y_i \sum_{i=1}^n \hat{y}_i}{\sqrt{n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2} \sqrt{n \sum_{i=1}^n \hat{y}_i^2 - (\sum_{i=1}^n \hat{y}_i)^2}} \quad (13)$$

Křížová validace

Je jedním z mnoha způsobů odhadu, jak moc dobře bude natrénovaný model (klasifikátor), pracovat na datech z testovací množiny.

Základní princip křížové validace

Počáteční datový soubor je nejprve náhodně rozdělen na podmnožiny (subsets) s přibližně stejným počtem záznamů v každé podmnožině. Dále je jedna podmnožina použita jako testovací oddíl, zatímco zbývající podmnožiny vykonávají roli tréninkového oddílu. Klasifikátor natrénuje model na trénovací množině a pomocí testovací množiny testuje přesnost a výkonnost tohoto modelu.

K- fold cross validation

Vstupní datový soubor je rozdělen v na k podmnožin. Tím je vytvořeno k modelů, které jsou všechny naučeny na různých podmnožinách počátečního datového souboru a kde každá z těchto podmnožin byla použita jako testovací oddíl přesně jednou [51].

Odhadovaná přesnost modelů pak může být vypočtena jako průměrná přesnost napříč k modely.

Volba počtu násobků, které se mají použít při křížové validaci k-fold, závisí na řadě faktorů. Především na počtu záznamů v počátečním datovém souboru [51]:

- Při práci s velmi řídkými datovými sadami může být výhodné vysoké k - až k bodu vynechání ($k = n$).
- Pokud má počáteční soubor dat vysoký počet záznamů, může být menší počet k docela přesný.
- Nejběžnější volbou pro k je 10. Obecně se zdá, že poskytuje nejlepší rovnováhu mezi časem zpracování a přesností.

Leave-one-out cross validation

Jedná se o zvláštní případ křížové validace, kdy k (počet záhybů) se rovná počtu záznamů v počátečním datovém souboru. Tato metoda je časově i výpočetně nejnáročnější, avšak udává nejlepší výsledky. Nejlépe využije pro datové sady s relativně nízkým počtem záznamů.

5 Analýza dat v rámci IoT

Analýza a zpracování dat ve spojení s IoT nabízí využití tzv. cloud computingu. V dnešní době jsou cloudové služby na vzestupu a stále častěji je využívají jak fyzické osoby, drobní podnikatelé a firmy, tak i nadnárodní společnosti (ty často samotné cloudy provozují). Vedle základních služeb, jako je obvyčejné cloudové uložení dat, spousta cloudových platforem nabízí nástroje pro analýzu dat a strojové učení. V této kapitole budou popsány možnosti strojového učení na dvou cloudových platformách – IBM Cloud a Microsoft Azure.

Z počátku bylo záměrem použití cloudové služby pro dlouhodobý výpočet predikce koncentrace oxidu uhličitýho v reálném čase. Postupem času při práci na obou cloudech se tento postup ukázal jako finančně náročný a pro tuto práci nepoužitelný. Finanční stránka každého projektu je důležitá, proto budu část této kapitoly též věnovat cenové politice obou cloudů.

Všechny predikce, vytvořené v této kapitole, se vztahují k datovému souboru, který byl naměřen v době od 25.6.2018 do 29.6.2018 na místnosti R203, perioda měření je jedna minuty, celý soubor čítá 7200 řádků.

Pipeline

V celém následujícím textu budu používat výraz "pipeline". Rád bych zde uvedl stručnou definici tohoto výrazu.

Pipeline je způsob, jak kodifikovat a automatizovat pracovní postup vedoucí k vytvoření modelu strojového učení. Pipeline strojového učení se skládá z několika postupných kroků, které dělají vše od extrakce a předzpracování dat až po trénink a nasazení modelu.

5.1 IBM Cloud

IBM Cloud je sada počítačových cloudových služeb provozovaných a vyvíjených společnostmi IBM. Cloud je hybridní a nabízí platformu se servisem (PaaS) a infrastrukturu jako službu (IaaS). Registrace je bezplatná a není k ní potřeba uvádět číslo platební karty. Pro studenty je dostupný bezplatný Education účet. Zprvu je tento účet platný na jeden rok, po roce je možné jej prodloužit.

IBM Cloud nabízí službu Watson studio, ta obsahuje řadu nástrojů v oblasti strojového a hlubokého učení. V rámci této práce jsem zkoumal možnosti třech nástrojů, jenž budou v následujících podkapitolách popsány.

5.1.1 SPSS Modeler

Jedná se o komplexní dataminingový nástroj, pokrývající celý proces od přípravy dat až po aplikaci výsledných modelů v praxi. Předností toho programu je široká nabídka nástrojů pro manipulaci s daty, modelovacích algoritmů a vizualizačních prostředků. Program je přehledně strukturovaný, má jednoduché a intuitivní prostředí založené na bázi drag-a-drop. K práci není nutnost psaní kódu, ale je zde možnost implementace skriptů.

Hlavní výhodou Watson studio je bezesporu desktopová verze SPSS Modeler, která využívá výpočetní výkon osobního počítače a její užívání není nijak časově omezeno (studentská licence je jen na jeden rok). Tuto verzi jsem používal při vytváření experimentů, kapitola 6.2..

Jednotlivé funkční bloky (nodes) jsou zde znázorněny v podobě geometrických tvarů (trojúhelníky, čtverce, pětiúhelníky). Funkční bloky se mezi sebou propojují a tím tok dat od jejich zdroje, až po analýzu výsledků predikce. Jednotlivé nástroje jsou rozděleny do sedmi záložek:

Source – V této záložce se nacházejí nástroje pro vkládání dat. Najdete zde nody pro vložení souboru z Microsoft Excel, flat files, SAS, soubory IBM SPSS Statistic (.sav) a také ODBC kompatibilní relační databáze. Tyto datové zdroje je možné mezi sebou kombinovat.

Record Operation – Tato záložka obsahuje celkem devět nodů. Ty se používají k provádění změn dat na úrovni záznamu. Tyto operace jsou důležité během fáze porozumění datům, tak i při přípravě dat či při získávání dat. Umožňují přizpůsobit data vašim konkrétním potřebám pro další zpracování.

Field operation – Jedná se o nejrozsáhlejší záložku obsahující sedmnáct různých nástrojů (nodů). Nástroje této záložky slouží pro vybírání, čištění či konstruování dat v rámci přípravy na analýzu. Pomocí těchto nástrojů je například možné vytváření atributů, překódování hodnot polí. Nejspíše nejčastěji využívaným nástrojem je node Type, který umožňuje přiřadit každé sadě dat úroveň měření, hodnotu a úlohu při modelování (input, target, atd.).

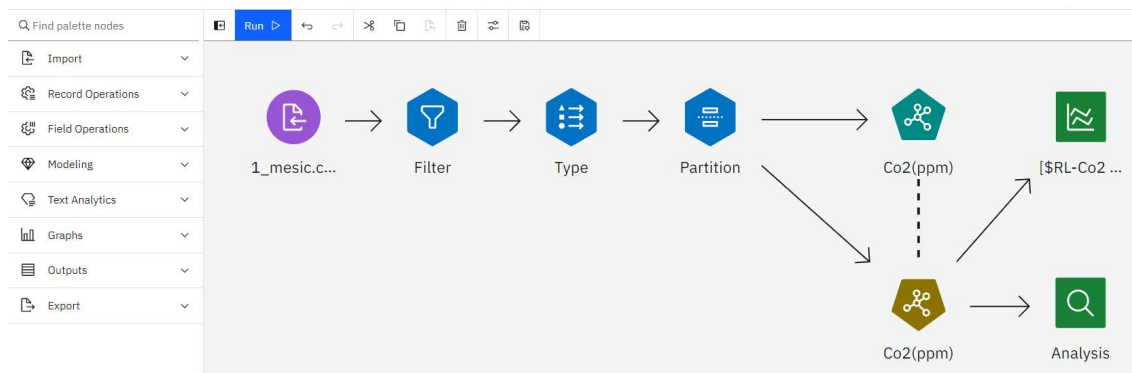
Graph – Záložka obsahující devět nodů. Nody jsou graficky ztvárněny trojúhelníky.

Modeling Outputs – Nástroje poskytující prostředky k získání informací o vašich datech a modelech. Nabízejí také mechanismus pro export dat v různých formátech do rozhraní s dalšími softwarovými nástroji.

Export – Poskytují nástroje pro export dat v různých formátech do rozhraní s dalšími softwarovými nástroji nebo pro finální uložení zpracovaných dat. Výstupy dat jsou možné ve formátech pro Microsoft Excel (.xls), Flat file, XML či SAS. Poté je možný výstup do databází a speciální formáty pro další analytické či statistické nástroje IBM.

Na obrázku níže je zobrazen pipeline pro výpočet predikce CO₂. Jedná se o prostředí cloudové verze SPSS. Podrobnější popis sestavení a nastavení pipeline je popsán v kapitole s experimenty (kap. 6.2.).

Modely vytvořené v tomto prostředí je možné ukládat, sdílet je a také nasazovat. Nutno zmínit, že k nasazování modely vytvořené v SPSS Modeler nejsou primárně určeny. Tento nástroj primárně slouží k provádění experimentů.



Obr. 9 Vytvoření pipeline pro predikci CO₂ nástroj SPSS Modeler

Na Obr.9 je sestrojený pipeline pro výpočet predikce koncentrace CO₂. Jako model je použit náhodný les a nastavenými pěti rozhodovacími stromy a maximální hloubkou stromu 10. Rozdělení vstupního souboru dat je následující: tréninková data (40 %), testovací data (30 %), validační data (30 %). Výsledkem experimentu jsou: LC: 0,984; MAE: 0,957.

5.1.2 AutoAI

Pro použití AutoAI není po uživateli vyžadována znalost programování, snad jen minimální znalost problematiky strojového učení. Tento nástroj, jak už název napovídá, pracuje zcela automaticky. Předzpracování dat, výběr predikčních metod nebo zhodnocení výsledků, vše dělá sám. Jediným úkonem uživatele je zvolení datového souboru, na kterém bude experiment proveden a zvolení cíle predikce. Soubor musí být ve formátu CSV.

Ke spuštění samotného experimentu je třeba na svém účtu vytvořit službu Machine Learning. Její instance se musí manuálně vložit do Watson studio. Výpočet experimentu trvá několik minut. Jeho výsledkem je seznam nejlépe ohodnocených predikčních modelů. Vybraný model je možné uložit a dále s ním pracovat nebo ho použít přímo k nasazení. Model lze uložit jako Model (lze ho implementovat do SPSS Modeler), nebo do formy notebooku. Ve formě notebooku je vygenerován kód v jazyce Python, který lze přímo v prostředí Watson otevřít a upravovat (Jupyter notebook).

Experiment vytvoří několik pipeline (počet lze nastavit), lišící se od sebe jak použitými predikčními modely a jejich různými nastaveními, tak i růzností předzpracování dat. Vytvořené pipeline nejdou editovat ani ukládat. Při uložení v podobě notebooku je možné vytvořený kód dodatečně upravovat, ale ani ten neobsahuje všechny prvky pipeline.

Rank	↑	Name	Algorithm	RMSE	Explained variat	Mean absolute ε	Mean squared e	Mean squared lc	Median absolutc	Root mean squa	R ²
★1		Pipeline 3	Extra Trees Regressor	2.478	0.984	0.686	6.140	0.000	0.000	0.004	0.984
2		Pipeline 4	Extra Trees Regressor	2.478	0.984	0.686	6.140	0.000	0.000	0.004	0.984
3		Pipeline 1	Extra Trees Regressor	2.713	0.981	0.759	7.389	0.000	0.000	0.005	0.981
4		Pipeline 2	Extra Trees Regressor	2.713	0.981	0.759	7.389	0.000	0.000	0.005	0.981
5		Pipeline 7	Random Forest Regressor	2.850	0.979	0.747	8.178	0.000	0.000	0.005	0.979
6		Pipeline 8	Random Forest Regressor	2.850	0.979	0.747	8.178	0.000	0.000	0.005	0.979
7		Pipeline 5	Random Forest Regressor	2.866	0.979	0.757	8.237	0.000	0.000	0.005	0.979
8		Pipeline 6	Random Forest Regressor	2.866	0.979	0.757	8.237	0.000	0.000	0.005	0.979

Obr. 10 Výsledek experimentu AutoAI

Na obrázku Obr. 10 je zobrazen výsledek experimentu, který byl zaměřen na výpočet predikce koncentrace CO₂. Výsledky jsou jednoznačné, na prvních osmi místech se umístili metody založené na regresních stromech (Random Forest, Extra Trees). Výsledky jsou velmi dobré, dosahují přesnosti predikce LC: 0,979; MAE: 0,747; MSE: 8,18; RMSE: 2,85. Výpočet experimentu trval 8 minut.

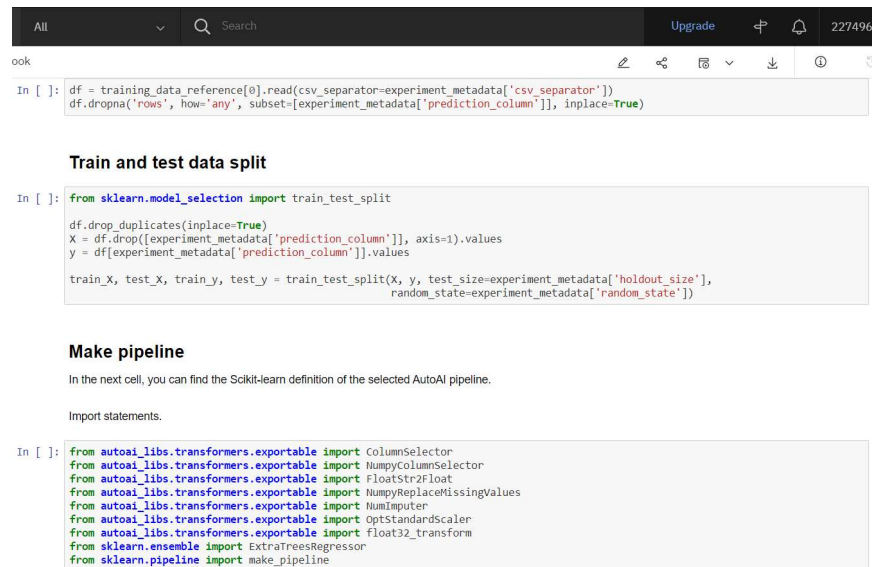
5.1.3 Notebook

Prostředí notebook nabízí jak psaní kódu v několika jazycích, tak obyčejné psaní poznámek v textové formě. Jsou zde podporovány jazyky:

- Python 3.7,
- R 3.6,
- Spark 3.0 & Scala/Python/R,
- Spark 2.4 & Scala/Python/R.

Pro použití notebook je možné zvolit požadovaný výpočetní výkon virtuálního zařízení.

V případě jazyku Python je prostředí shodné s prostředím Jupyter Notebook. Kód je psán do oddělených buněk. Buňky lze samostatně spouštět. Toto prostředí umožňuje automatické generování kódu, kdy například při importu dat do kódu stačí najít složku na cloudovém uložení a část kódu pro import dat se vytvoří sama.



```
In [ ]: df = training_data_reference[0].read(csv_separator=experiment_metadata['csv_separator'])
df.dropna('rows', how='any', subset=[experiment_metadata['prediction_column']], inplace=True)

Train and test data split

In [ ]: from sklearn.model_selection import train_test_split

df.drop_duplicates(inplace=True)
X = df.drop([experiment_metadata['prediction_column']], axis=1).values
y = df[experiment_metadata['prediction_column']].values

train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=experiment_metadata['holdout_size'],
                                                    random_state=experiment_metadata['random_state'])

Make pipeline

In the next cell, you can find the Scikit-learn definition of the selected AutoAI pipeline.

Import statements.

In [ ]: from autoai_libs.transformers.exportable import ColumnSelector
from autoai_libs.transformers.exportable import NumpyColumnSelector
from autoai_libs.transformers.exportable import FloatStr2Float
from autoai_libs.transformers.exportable import NumpyReplaceMissingValues
from autoai_libs.transformers.exportable import NumImputer
from autoai_libs.transformers.exportable import OptStandardScaler
from autoai_libs.transformers.exportable import Float32_transform
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.pipeline import make_pipeline
```

Obr. 11 Ukázka prostředí Notebook Python 3.7

5.1.4 Nasazení modelů

Jedná se o funkci, skrze kterou lze nasazovat vytvořené modely. DeploySpace je samostatná součást Watson studia a je nutné ji vytvořit zvlášť, mimo projekt, ve kterém se modely vytváří. K jeho používání je nutné vytvořit službu Machine Learning.

Nasazování modelů je proces, jenž vede ke zpřístupnění vytvořených modelů aplikacím třetích stran. Ve chvíli, kdy je model nasazen, ale na něm provádět predikci dat v reálném čase. Vytvořený model, pipeline, experiment je přístupný pouze v prostředí Watson studio a nelze ho automaticky spouštět. Pokud je model nasazen, jsou mu přiřazeny identifikační atributy a je možné k němu přistupovat skrze jeho vlastní API, stává se webovým servisem.

Prvním krokem nasazení modelu je vytvoření samotného Deployment space. Ten se vytváří v záložce "Deployments", nacházející se na levé straně úvodní obrazovky Watson studio. Po vytvoření prostoru pro nasazení je možné jednotlivé modely do tohoto prostoru přesunout. Zde se modely mohou aktivně nasadit. Je třeba si dát pozor, že od chvíle, kdy je model aktivně nasazen, se započítává cena za jeho nasazení. To i ve chvíli, kdy model nepoužíváte. jeho nasazení nelze nijak stopnou. Jedinou možností je jeho úplné smazání z Deployments. V Deployments je možné vytvořené nasazení modelu otestovat.

5.1.5 Cenová politika cloudu

V tomto odstavci bude popsáno, jakým způsobem je bezplatný účet limitovaný a jakým způsobem je určována cena za jednotlivé nástroje. Celé je to postaveno na jednotce CUH (Capacity Unit Hours). Služba Watson studio má na studentský účet měsíční dotaci 50 CUH. Po vyčerpání je služba nepřístupná, dokud není proveden upgrade účtu na zpoplatněnou verzi nebo dokud nezačne další měsíc. To, kolik jaký nástroj spotřebovává CUH, závisí na použitém výkonu virtuálního zařízení. Pro nástroj SPSS Modeler je defaultně nastavený výkon 4 vCPU, 16 GB RAM, který stojí 2 CUH za hodinu užívání. V tabulce je uveden přehled cen služeb.

Tab. 1 Tabulka časové dotace služb IBM Cloud

Služba	Časová dotace
Watson studio	50 CUH
Machine Learning	20 CUH

Tab. 2 Ceny za použití virtuálních výkonů zařízení

Nástroj	Výkon virtuálního zařízení	Cena
SPSS Modeler	4 vCPU, 16 GB RAM	2 CUH
AutoAI	8 vCPU, 32 GB RAM	20 CUH
Notebook – Python 3.7	1 vCPU, 4 GB RAM	0.5 CUH
	2 vCPU, 8 GB RAM	1 CUH
	4 vCPU, 16 GB RAM	2 CUH

V době, kdy jsem tuto kapitolu psal (2.4.2021), byla cena jedné CUH 0.42 dolaru.

5.2 Microsoft Azure

Azure je cloudová platforma společnosti Microsoft. Azure nabízí širokou paletu vzájemně se doplňujících služeb. V rámci této platformy jsou dostupné nástroje výpočetní kapacity, úložiště, síťové funkce a aplikace. Jedná se o hybridní cloud, ve kterém je možné provozovat PaaS (Platform as a Service), IaaS (Infrastructure as a Service) i SaaS (Software as a Service).

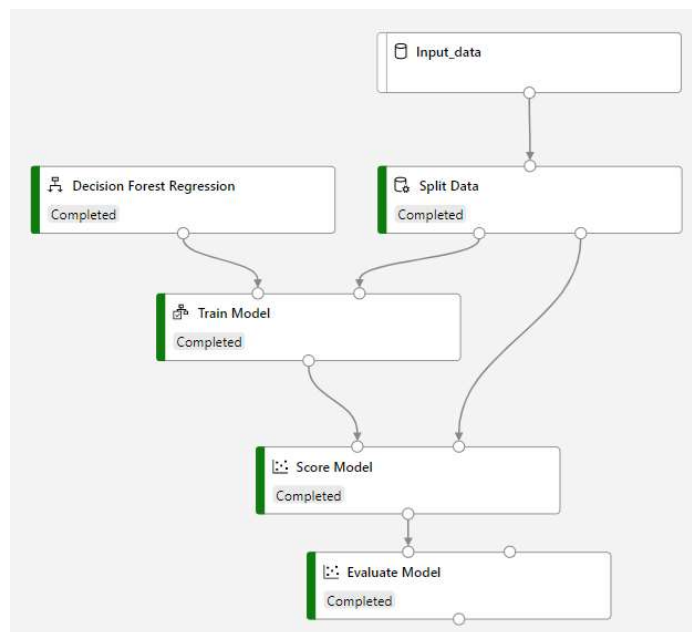
V této práci jsou použity především služby výpočetní kapacity, přesněji nástroj Azure Machine Learning studio, ve kterém jsou dostupné nástroje strojového učení.

5.2.1 Designer

Nástroj pro vytváření, učení a experimenty s modely strojového učení. Dalo by se říci, že je alternativou SPSS Modeler. Je založen na drag-n-drop prostředí, takže vytvoření pipeline je velmi jednoduché a intuitivní. Nástroj po uživateli nevyžaduje žádné znalosti programování. V Designer je nutná alespoň základní znalost problematiky strojového učení, práce s datovými soubory a schopnost vyhodnotit výsledky vytvořených modelů.

Práce v nástroji Designer by se dala shrnout třemi kroky. V prvním kroku je do prostředí vložen datový vstup (uložená data přímo z cloud, CSV soubor z osobního počítače či data z nějakého připojeného zařízení). Vložená data jsou dle potřeby předzpracována. V druhém kroku je do prostředí přidán node pro trénování modelu a samotný model. V posledním kroku jsou přidány nody pro vyhodnocení výsledků vytvořeného modelu.

Přímo z prostředí Designer je možné vytvořený a naučený pipeline nasadit jako webovou službu.



Obr. 12 Pipeline vytvoření v nástroji Designer

Na obrázku 12 je zobrazen sestavený pipeline pro výpočet predikce koncentrace CO₂. Jako model je zde náhodný les s pěti regresními stromy a maximální hloubkou stromu 10. Rozdělení datového souboru je 80 % trénovací, 20 % testovací. Výsledky naučeného testování naučeného modelu jsou velmi dobré. r^2 : 0,965; MAE: 0,078; RMSE: 0,840.

5.2.2 Automated ML

Též často označované jako automatizované ML nebo AutoML, je proces časově náročného zpracování interaktivních úloh vývoje modelů strojového učení. Nabízí velmi jednoduchý a rychlý vývoj modelu strojového učení. Pro použití tohoto nástroje není nutná znalost programování a pouze minimální znalost problematiky strojového učení. Kroky jako je předzpracování dat, výběr modelovacích technik i zhodnocení výsledků experimentu provádí automaticky nástroj sám. Na uživateli je pouze vložení vstupních dat a základní nastavení v podobě zvolení cíle predikce a rozhodnutí, zda se jedná o klasifikaci nebo regresi.

Pro zahájení experimentu je nutné mít vytvořenou instanci výpočetního výkonu, které chceme využívat. Po zahájení experimentu probíhá příprava, které trvá 10 až 15 minut. Jakmile se výpočet rozběhne, trvá každá iterace 2 až 3 minuty. Celkově tak může čas výpočtu experimentu trvat minimálně půl hodiny.

Experiment, provedený v rámci predikce CO₂, trval hodinu a pět minut. Výsledkem je 49 různých pipeline, které se od sebe liší použitým predikčním modelem, tak použitou technikou předzpracování dat. Jedním z nejlépe hodnocených pipeline je sloužení modelu Random Forest a škálování dat prostřednictvím MaxAbsScaler. Výsledky tohoto modelu jsou: r^2 : 0,979; MAE: 0,821; RMSE: 2,845;. Vytvořené pipeline v rámci experimentu je možné bez dalších úprav nasadit.

Algorithm name	Explained	Spearman correlation
StackEnsemble	View explanation	0.99041
VotingEnsemble		0.99021
StandardScalerWrapper, XGBoostRegressor		0.98970
MaxAbsScaler, LightGBM		0.98831
MinMaxScaler, RandomForest		0.98716
SparseNormalizer, XGBoostRegressor		0.98659
SparseNormalizer, XGBoostRegressor		0.98564
StandardScalerWrapper, XGBoostRegressor		0.98477
MaxAbsScaler, RandomForest		0.98395
StandardScalerWrapper, LightGBM		0.98352
SparseNormalizer, XGBoostRegressor		0.98176

Obr. 13 Výsledky experimentu v nástroji Automated ML

5.2.3 Notebook

Práce s notebookem nabízí v podstatě nejširší nabídku ze všech nabízených nástrojů. Jeho možnosti jsou limitovány znalostmi programátora. Prostřednictvím knihovny Azure SDK je možné pracovat s prostředím Machine Learning studio z jakéhokoli IDE.

Podporované formáty:

- Notebook Python/Jupyter notebook (.ipynb),
- Python (.py),
- R (.R),
- Bash (.sh),
- Text (.txt).

Prostředí pro psaní kódu v Notebook s formátem .ipynb je velmi podobné prostředí Jupyter. Kód je zde možné psát oddělených blocích, ty lze jednotlivě spouštět.

```
[ ] 1 green_taxi_dataset = Dataset.Tabular.from_parquet_files(path="https://automlsamplenotebookdata.blob.core.windows.net/automl-sample-
2 green_taxi_df = green_taxi_dataset.to_pandas_dataframe()
3 green_taxi_df.head(10)
```

Now that the initial data is loaded, define a function to create various time-based features from the pickup datetime field. This will create new fields for the month number, day of month, day of week, and hour of day, and will allow the model to factor in time-based seasonality.

Use the `apply()` function on the dataframe to iteratively apply the `build_time_features()` function to each row in the taxi data.

```
[ ] 1 def build_time_features(vector):
2 pickup_datetime = vector[0]
3 month_num = pickup_datetime.month
4 day_of_month = pickup_datetime.day
5 day_of_week = pickup_datetime.weekday()
6 hour_of_day = pickup_datetime.hour
7
8 return pd.Series([month_num, day_of_month, day_of_week, hour_of_day])
9
10 green_taxi_df[["month_num", "day_of_month", "day_of_week", "hour_of_day"]] = green_taxi_df[["lpep_pickup_datetime"]].apply(build_time_
11 green_taxi_df.head(10)
```

Obr. 14 Příklad části kódu v Notebook

5.2.4 Nasazení modelů

Služba Machine Learning studio umožňuje nasazovat vytvořené modely strojového učení jako webové servisy. Na rozdíl od IBM Cloud zde není pro nasazování modelů nutné použít další službu. Lze si zde zvolit z instancí výpočetních výkonů, který má být na nasazení modelu použit. Další výhodou oproti IBM je možnost nasazení modelu zastavit. Při tomto zastavení se přestane započítávat cena za použitý virtuální výpočetní výkon, ale nedojde ke smazání atributů, definující připojení k modelu.

5.2.5 Cenová politika cloudu

Na tomto cloudu je cenová politika velmi jednoduchá, platí za to, co používáte. Při registraci k bezplatnému účtu dostane uživatel na svůj účet kredit 100 dolarů. Za každou službu, kterou v cloudu využívá, platí a platby jsou strhávány z jeho kreditu. Ve chvíli, kdy je kredit vyčerpán, jsou všechny služby pozastaveny do doby, kdy je do účtu vložena platební karta a účet je upgradován na placenou verzi.

Tab. 3 Ceny výpočetních výkonů virtuálního zařízení

Výpočetní výkon	Cena
2 vCPU, 7 GB RAM	0,13/hodinu
4 vCPU, 8 GB RAM	0,19/hodinu
4 vCPU, 14 GB RAM	0,26/hodinu

5.3 Zhodnocení práce s cloudy

Watson studio je pro práci se strojovým učáním opravdu silným nástrojem. Výpočty i sestavení pipeline v SPSS Modeler jsou velmi rychlé a práce s ním je snadná a intuitivní. AutoAI je skvělým nástrojem pro rychlý vytvoření experimentů. Největším plusem Watson studio je desktopová verze SPSS Modeler, která není nijak limitovaná (pro studentský účet je bezplatná licence na jeden rok, jinak je trial verze na jeden měsíc). Největším mínusem Watson studio je jeho měsíční časová dotace. Ta používání této služby omezuje pouze na seznámení se s ní. Další zápornou stránkou je cena služeb IBM, především za nasazení vytvořených modelů v reálném čase. Týdenní výpočet predikce CO₂ by zde stál okolo 500 dolarů, což je nepřipustné.

Práce v Azure je oproti IBM velmi pomalá. Výpočet experimentu v Designer trvá z pravidla několik minut. Výpočet experimentu Automated ML trvá přes hodinu. Vytváření instancí nástrojů nebo výpočetních výkonů též trvá několik minut. rychlost výpočtu nemá žádný vliv na přesnost vytvořených pipeline. Celkově má Azure velmi dobře zpracovanou podporu ke všem svým službám a jejich nástrojům. Po dvou týdnech používání Azure jsem jich 25 % svého kreditu vyčerpal. Téměř 20 dolarů bylo za poplatky za vytvořená uložení, bez kterých nejde Machine Learning studio používat. Pokud bych použil pro výpočet predikce nejnižší dostupný výpočetní výkon, stál by týden výpočtů cca 30 dolarů.

Tyto cloudové platformy i jednotlivé nástroje používají různá kritéria, proto není možné je zcela jednoznačně seřadit a mezi sebou hodnotit. Z experimentů provedených v AutoAI a Automated ML byly na vrcholu žebříčku metody strojového učení založené na regresních stromech, především metody náhodná les a ExtraTrees.

6 Experimenty

V teoretické části této práce se nachází popisy tří matematických metod, které mohou být použity pro predikci CO₂. Výběr těchto metod je obsažen v části této kapitoly. Také zde budou provedeny experimenty s těmito metodami. Experimenty jsou vytvořeny v programu SPSS Modeler.

6.1 Výběr metod

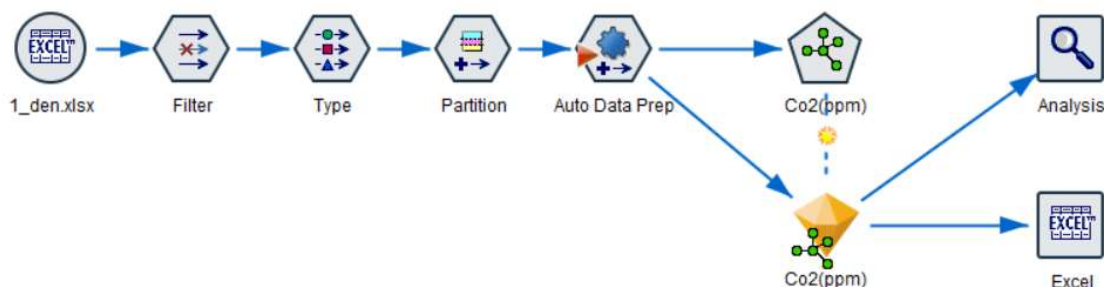
Pro výběr vhodné matematické metody disponuje SPSS Modeler funkcí, která na základě vlastností vstupních dat sestaví žebříček nejpřesnějších predikčních metod, které software nabízí. Výběr zohledňuje pouze kritériu lineární korelace. Na třech místech se umístili metody náhodný les, ExtraTrees a CHAID. Náhodný les a extra trees jsou oboje založeny na binárních regresních stromech, zatím co CHAID je zažen na nebinárních regresních stromech. Vzhledem k rozmatitosti tedy volím použití metod náhodný les, CHAID a pro zástce hlubokého učení neuronové sítě.

6.2 Experimenty

Prvním experimentem je nalezení vhodného nastavení parametrů pro vytvoření predikčního modelu. Zde budu hledat vhodné nastavení parametru každé z metod. K tomu je použit datový soubor s délkou naměřeného intervalu jeden týden.

Druhým experimentem je zjištění vlivu délky intervalu vstupních dat na přesnosti predikce. Zde jsou porovnány tři délky datových souborů: den (144 hodnot), týden (1152 hodnot), měsíc (4032 hodnot). Každá délka intervalu naměřených dat má dva různé datové soubory. Perioda vzorkování naměřených dat je 10 minut. Všechny datové soubory použité při těchto experimentech mají normalizované hodnoty.

Vytvoření modelu v SPSS Modeler je vyobrazeno na Obr. 15. Toto zapojení je shodné pro všechny metody, liší se pouze v použitém modelu. Vstupní datový soubor je nahrán z excelového souboru prostřednictvím Excel SourceNode. Dále jsou pomocí node Filter odstraněny z dalšího zpracování nepotřebná data. TypeNode zajišťuje přiřazení role pro každé pole ze souboru vstupních dat. V tomto případě je pole dat s hodnotami Co2 označeno jako *Target*, pole s daty má roli *RecordingID* a nakonec pole dat s hodnotami vnitřní, venkovní teploty a relativní vlhkosti mají roli *Input*. Následně jsou takto označená data rozdělena v PartitionNode do tří souborů dat: tréninková data (40 %), testovací data (30 %), validační data (30 %). Tréninková data jsou použita pro vytvoření samotného modelu, testovací data se využívají při vytváření modelu ke zjištění přesnosti vytvořeného modelu, validační data slouží ke konečnému ověření a zjištění skutečné přesnosti predikce vytvořeného modelu. Node AutoDataPrep je zde využit pro předzpracování dat před samotným vytvořením modelu. Tento node je zaměřen na zvýšení přesnosti predikčního modelu. Nakonec se dostáváme k node s pětistranným tvarem, který představuje predikční metodu. V tomto node se provádí veškeré nastavení pro vytvoření modelu dané metody. Po spuštění simulace je vytvořen node ve tvaru diamantu. K tomuto node se poté připojují různé nody pro výstup dat, grafy a nástroje pro analýzu vypočtených dat.



Obr. 15 Zapojení pipeline v SPSS Modeler, model neuronové sítě

Náhodný les

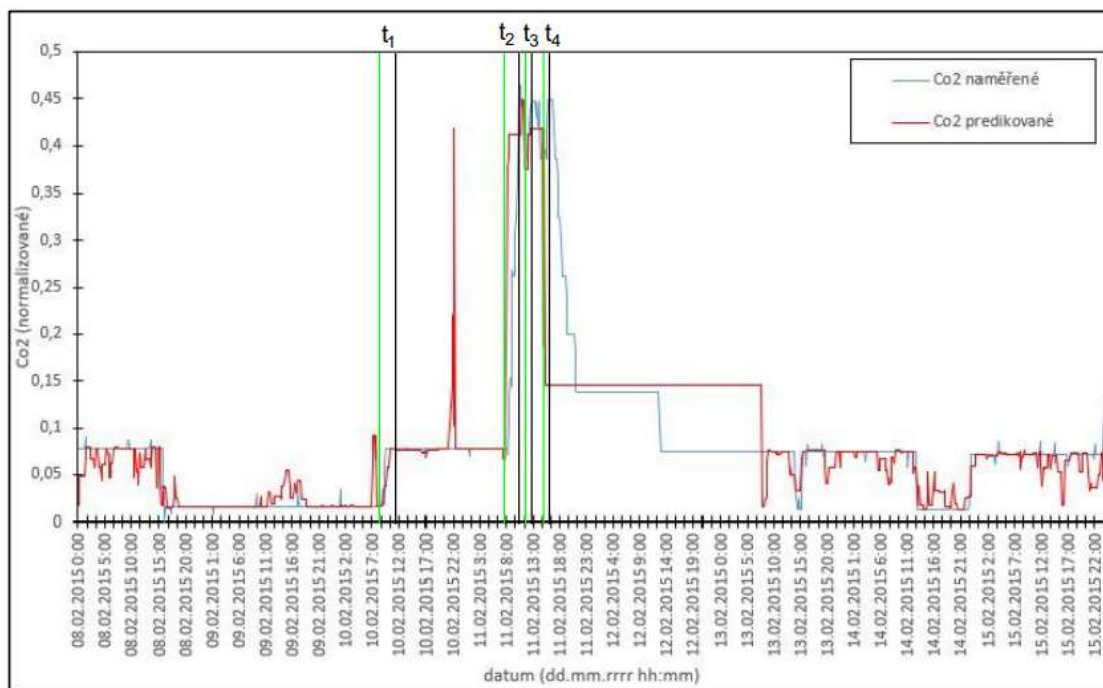
U metody náhodného lesu sleduji vliv počtu vytvořených stromů modelu na přesnost výsledku. Hodnoty kritérií pro různé hodnoty počtu vytvořených stromů jsou uvedeny v Tab. 4. U této metody ideální nastavení nelze naprosto přesně určit, jelikož výsledky této metody se liší v závislosti na opakování. Dva výpočty predikce CO₂ se stejným vstupním datovým souborem a stejným nastavením nejsou stejné, jsou si však velmi podobné. Pokud budeme uvažovat nejlepší výsledek z Tab. 4, jednalo by se o nastavení dvou rozhodovacích stromů (LC: 0,746; MAE: 0,020; MSE: 0,0021; MAPE: 25,22 %).

Z výsledků tohoto experimentu a zkušeností, které jsem při tomto experimentu o této metodě získal, volím jako vhodné nastavení této metody použití pěti rozhodovacích stromů. Tuto volbu podporuje několik faktů. Prvně, jak je vidět v tabulce Tab. 4, výsledky kritérií při zvyšování počtu vytvořených rozhodovacích stromů kolísají, jednou se zlepší, při dalším kroku se opět zhorší (např. mezi pěti a šesti stromy je značné zlepšení přesnosti predikce, ovšem mezi šesti a sedmi stromy přesnost klesla). Dalším faktem je náročnost a doba výpočtu predikce při použití vysokého počtu stromů. Výsledky při pěti stromech jsou srovnatelné s výsledky při použití deseti i více stromů.

Tab. 4 Hodnoty kritérií pro různá nastavení počtu vytvořených rozhodovacích stromů v modelu náhodného lesu

Počet stromů	LC	MAE	MSE	MAPE (%)
20	0,757	0,021	0,0020	28,91
15	0,767	0,019	0,0019	26,06
10	0,771	0,021	0,0020	30,64
9	0,759	0,021	0,0020	30,65
8	0,775	0,019	0,0020	29,45
7	0,757	0,021	0,0020	28,38
6	0,778	0,019	0,0020	28,16
5	0,734	0,021	0,0022	28,96
4	0,759	0,020	0,0020	30,31
3	0,753	0,020	0,0022	29,33
2	0,746	0,020	0,0021	25,22
1	0,677	0,025	0,0029	30,41

Srovnání predikovaných a naměřených dat v grafickém podání se ukazují nedostatky predikce. V určitých časech (8.2.2020 2:40 – 8.2.2020 14:15) se vyskytují výkyvy hodnot CO₂. Větším problémem jsou peaky v časech 8.2.2020 18:10, 10.2.2020 7:10 a 10.2.2020 22:20. Ty se dají jednoduše splést se vzrůstem hodnot CO₂ při přítomnosti osob v místnosti. Důležitým bodem je přesné predikování strmého a velkého nárůstu koncentrace CO₂ v čase 11.1.2020 7:30 (Obr. 16 – t_{2.1}), kdy na učeňně nejspíše začala výuka a vyskytovalo se v ní více osob.



Obr. 16 Graf porovnání naměřených a predikovaných hodnot CO₂. Modelem náhodného lesu, použito pět rozhodovacích stromů. Datový soubor 8.2.2020 – 15.2.2020. Zelená svislá čára značí příchod osoby do místnosti, černá svislá čára značí odchod. Příchod t₁ = 10.2.2020 08:50, odchod t₁ = 10.2.2020 09:40, délka pobytu Δt₁ = 00:50. Příchod t₂ = 11.2.2020 07:30, odchod t₂ = 11.2.2020 10:10, délka pobytu Δt₂ = 02:40. Příchod t₃ = 11.2.2020 12:00, odchod t₃ = 11.2.2020 12:50, délka pobytu Δt₃ = 00:50. Příchod t₄ = 11.2.2020 15:20, odchod t₄ = 11.2.2020 15:40, délka pobytu Δt₄ = 00:20.

Z výsledků srovnání délek intervalů (Tab. 5) vstupních souborů dat je u této metody viditelný nárůst chyby predikce s narůstající velikostí souboru dat. Nejlépe vychází délka intervalu dodaných dat o délce jednoho dne. Naprosto nejlépe vyšel soubor jednoho dne z 8.2.2015 (LC: 0,966; MAE: 0,002; MSE: 7,23 · 10⁻⁵; MAPE: 5,64 %).

Tab. 5 Hodnoty kritérií pro různé délky intervalu vstupního datového souboru, model náhodného lesu

Datový soubor	LC	MAE	MSE	MAPE (%)
1.2. – 28.2.2015	0,723	0,022	0,0024	23,21
1.6. – 28.6.2015	0,780	0,036	0,0038	22,12
8.2. – 15.2.2015	0,867	0,021	0,0021	27,04
8.6. – 15.6.2015	0,892	0,035	0,0050	12,75
8.2.2015	0,966	0,002	7,23 · 10 ⁻⁵	5,64
8.6.2015	0,905	0,045	0,068	17,71

CHAID

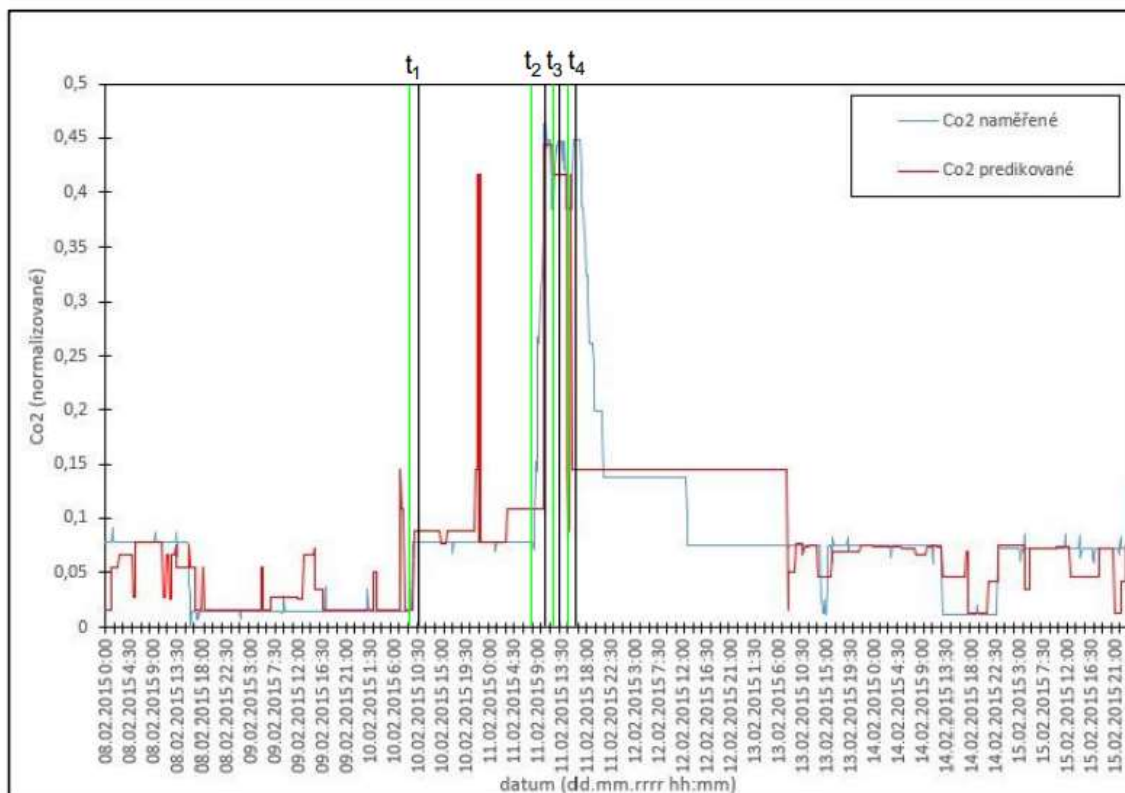
U metody CHAID hledám ideální nastavení parametru maximální hloubky vytvořeného stromu. Možné nastavení této metody je rozsáhlé, obsahuje 9 různých parametrů.

Výsledky přesnosti predikce hodnot CO_2 jsou zaznamenány v tabulce Tab. 6. Největší změnou v přesnosti predikce u metody CHAID je mezi vytvořením jedné a dvou vrstev rozhodovacího stromů. Rozdíl mezi použitím dvou a více vrstev není už tak značný, přesto je podstatný. Nejlepších výsledků je dosaženo při použití čtyř vrstev hloubky stromu (LC: 0,770; MAE: 0,022; MSE: 0,0026; MAPE 43,35 %). Od této hodnoty se přesnosti predikce nijak nemění. Jako vhodné nastavení pro model CHAID volím použití šesti vrstev hloubky stromu. Tato hodnota představuje určitou rezervu i pro větší datové soubory, zároveň se nejedná o předimenzování tohoto nastavení. Výhodou této metody je nezávislost jejího výsledku na opakování.

Tab. 6 Hodnoty kritérií pro různá nastavení maximální hloubky vytvořeného stromu modelu CHAID

Maximální hloubka stromu	LC	MAE	MSE	MAPE (%)
10	0,770	0,022	0,0026	43,35
6	0,770	0,022	0,0026	43,35
5	0,770	0,022	0,0026	43,35
4	0,770	0,022	0,0026	43,35
3	0,769	0,023	0,0026	43,75
2	0,707	0,028	0,0035	54,56
1	0,557	0,038	0,0047	91,68

Z grafického porovnání predikovaných a naměřených dat (Obr. 17) je vidět spousta výkyvů a peaků, které falešně signalizují přítomnost osob v místnosti (8.2.2020 17:30, 9.2.2020 5:20, 9.2.2020 7:40, 10.2.2020 1:50, 10.2.2020 20:20). Celkově se predikovaný průběh v podstatných bodech shoduje v naměřeném, obsahuje však spoustu míst, které se tváří jako přítomnost osob, i když tomu tak není.



Obr. 17 Graf porovnání naměřených a predikovaných hodnot CO_2 . Model CHAID, použito šest vrstev rozhodovacího stromu. Datový soubor 8.2.2020 – 15.2.2020. Zelená svislá čára značí příchod osoby do místnosti, černá svislá čára značí odchod. Příchod $t_1 = 10.2.2020\ 08:50$, odchod $t_1 = 10.2.2020\ 9:40$, délka pobytu $\Delta t_1 = 00:50$. Příchod $t_2 = 11.2.2020\ 07:30$, odchod $t_2 = 11.2.2020\ 10:10$, délka pobytu $\Delta t_2 = 02:40$. Příchod $t_3 = 11.2.2020\ 12:00$, odchod $t_3 = 11.2.2020\ 12:50$, délka pobytu $\Delta t_3 = 00:50$. Příchod $t_4 = 11.2.2020\ 15:20$, odchod $t_4 = 11.2.2020\ 15:40$, délka pobytu $\Delta t_4 = 00:20$.

V tabulce Tab. 7 jsou uvedeny hodnoty kritérií predikce pro různé délky datového souboru při nastavení hloubky vytvořeného stromu šest vrstev. Srovnání přesnosti predikce při změně délky intervalu naměřených dat nejlépe vychází datové soubory naměřené během jednoho dne, konkrétně 8.2.2015 (LC: 0,991; MAE: 0,002; MSE: $9,75 \cdot 10^{-6}$; MAPE 3,33 %). Chyba predikce roste přímou úměrou, při zvětšující se délce intervalu dodaných dat se zvětšuje chyba predikce.

Tab. 7 Hodnoty kritérií pro různé délky intervalu vstupního datového souboru, model CHAID

Datový soubor	LC	MAE	MSE	MAPE (%)
1.2. – 28.2.2015	0,458	0,041	0,0053	66,40
1.6. – 28.6.2015	0,533	0,061	0,0094	53,60
8.2. – 15.2.2015	0,770	0,022	0,0026	43,35
8.6. – 15.6.2015	0,745	0,066	0,0146	31,23
8.2.2015	0,991	0,002	$9,75 \cdot 10^{-6}$	3,33
8.6.2015	0,902	0,043	0,0071	12,56

Neuronové sítě

U neuronových sítí hledám ideální nastavení počtů neuronů ve vnitřních vrstvách modelu neuronové sítě. Nalezení ideálního počtu neuronů pro maximální přesnost predikce je velmi obtížné, snad i téměř

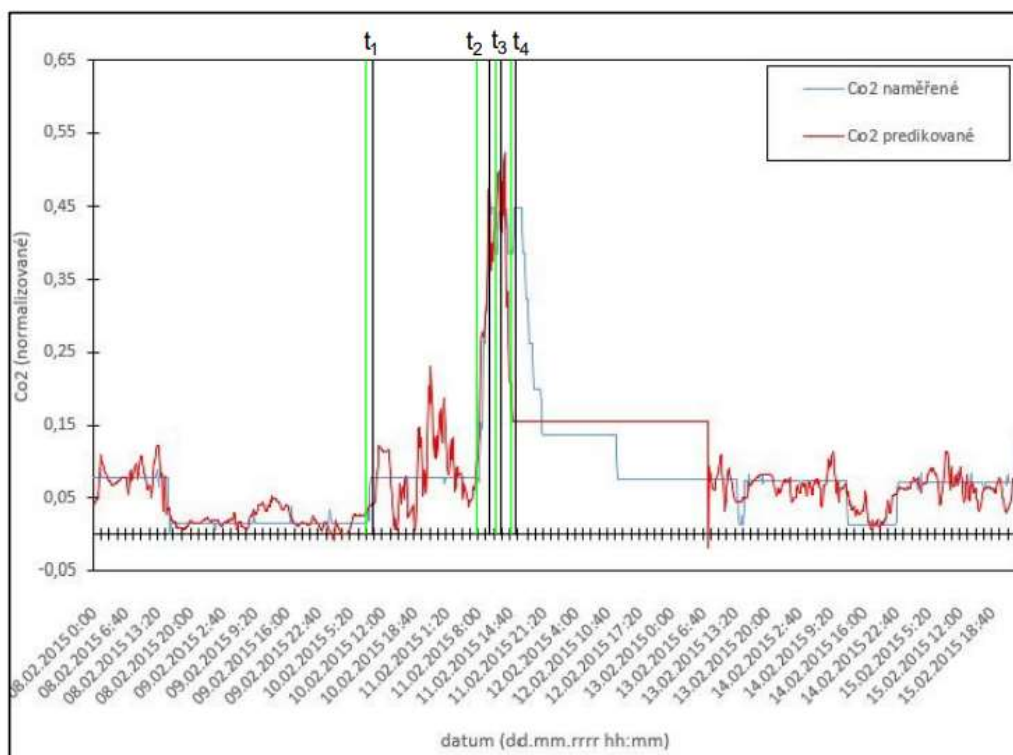
nemožné. Pokud vezmeme v potaz možnost nastavení dvou vnitřních vrstev neuronové sítě, přičemž každá může nabývat hodnot od nuly do například 500 (možnost nastavení v SPSS Modeler je až do jednotek tisíc) dostáváme 250 000 možností nastavení. Při vytváření toho experimentu jsem si všiml skutečnosti, že ideální nastavení pro jeden datový soubor, není ideální pro všechny datové soubory. Pokud bychom našli ideální nastavení na jednom datovém souboru, který popisuje určitý průběh veličiny CO₂ s určitým počtem hodnot, pro jiný datový soubor, byť pouze s jiným průběhem veličiny CO₂ ale stejným počtem hodnot, bude ideální nastavení opět jiné. Avšak i různé počty neuronů v jednotlivých vrstvách dosahují v určitých případech velmi podobných výsledků. Vzhledem k těmto poznatkům budu hledat pouze ideální nastavení datové soubory naměřené v intervalu jednoho týdne od 8.2.2015 do 15.2.2015.

Pro tento datový soubor nejlépe vychází nastavení neuronů ve vnitřních vrstvách 200 – 100 (LC: 0,736; MAE: 0,028; MSE: 0,0024; MAPE 46,78 %). Výsledky se při různých nastaveních příliš nemění.

Tab. 8 Hodnoty kritérií pro různé počty neuronů ve vnořených vrstvách modelu neuronové sítě

Počty neuronů	LC	MAE	MSE	MAPE (%)
200-100	0,736	0,028	0,0024	46,78
100-100	0,734	0,029	0,0025	51,88
50-100	0,728	0,030	0,0025	54,84
200-80	0,736	0,029	0,0025	50,90
150-50	0,737	0,029	0,0025	52,76
50-30	0,721	0,030	0,0026	64,38
20-20	0,717	0,031	0,0026	61,63

Na grafickém srovnání naměřených a predikovaných dat (Obr. 18) je jasně viditelná roztřesenost průběhu predikovaných dat v celém časovém rozsahu. Predikovaný průběh kopíruje průběh naměřený jen omezeně. V místech, kdy jsou naměřené hodnoty konstantní, vykazuje predikovaný průběh různé poklesy a zvýšení. Tyto rozdíly v predikovaném a vypočteném průběhu je možné sledovat u všech délek datových souborů.



Obr. 18 Graf porovnání naměřených a predikovaných hodnot CO_2 . Modelem neuronových sítí, použito 200 neuronů v první vnořené vrstvě, 100 neuronů ve vrstvě druhé. Datový soubor 8.2.2020 – 15.2.2020. Zelená svislá čára značí příchod osoby do místnosti, černá svislá čára značí odchod. Příchod $t_1 = 10.2.2020\ 08:50$, odchod $t_1 = 10.2.2020\ 9:40$, délka pobytu $\Delta t_1 = 00:50$. Příchod $t_2 = 11.2.2020\ 07:30$, odchod $t_2 = 11.2.2020\ 10:10$, délka pobytu $\Delta t_2 = 02:40$. Příchod $t_3 = 11.2.2020\ 12:00$, odchod $t_3 = 11.2.2020\ 12:50$, délka pobytu $\Delta t_3 = 00:50$. Příchod $t_4 = 11.2.2020\ 15:20$, odchod $t_4 = 11.2.2020\ 15:40$, délka pobytu $\Delta t_4 = 00:20$.

Výsledky experimentu, který zkoumá vliv délky intervalu dodaného souboru dat jsou zapsány v tabulce Tab. 9. Z pohledu délky intervalu naměřených dat vychází nejlépe délka jednoho dne. Opět nejlépe vyšel den 8.2.2015 (LC: 0,911; MAE: 0,007; MSE: $7,7890 \cdot 10^{-5}$; MAPE: 12,33 %). Stejně jako u předchozích metod můžeme sledovat postupné zhoršování přesnosti predikce při vzrůstajícím intervalu dodaných naměřených dat.

Tab. 9 Hodnoty kritérií pro různé délky intervalu vstupního datového souboru, model neuronové sítě

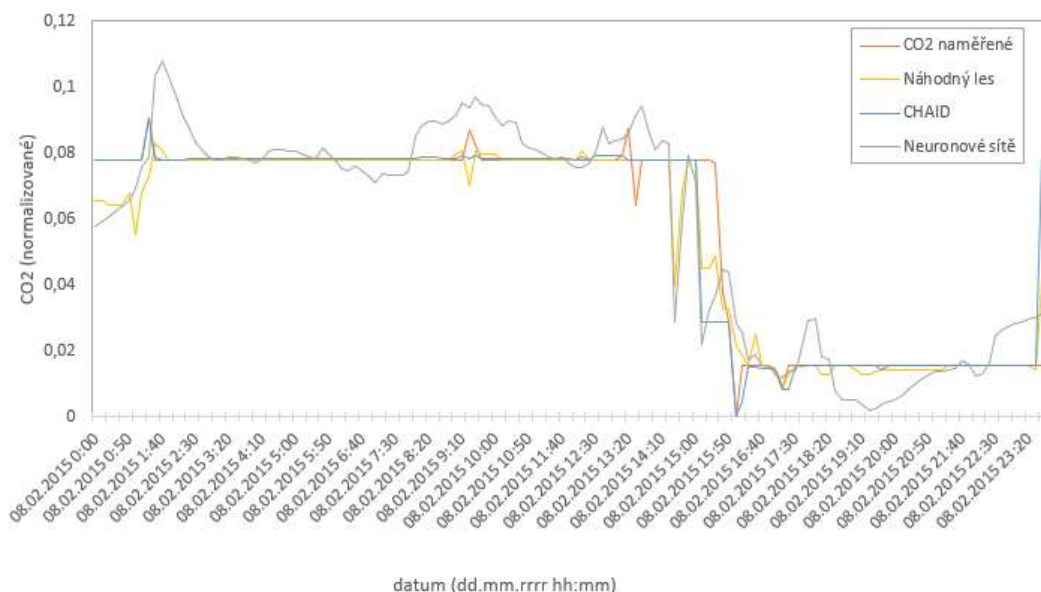
Datový soubor	LC	MAE	MSE	MAPE (%)
1.2. – 28.2.2015	0,425	0,044	0,0056	71,50
1.6. – 28.6.2015	0,502	0,067	0,0100	60,29
8.2. – 15.2.2015	0,734	0,029	0,0025	51,88
8.6. – 15.6.2015	0,852	0,057	0,0088	35,02
8.2.2015	0,911	0,007	$7,7890 \cdot 10^{-5}$	12,33
8.6.2015	0,880	0,034	0,0087	30,52

Porovnání predikovaných průběhů pro různé délky datového souboru

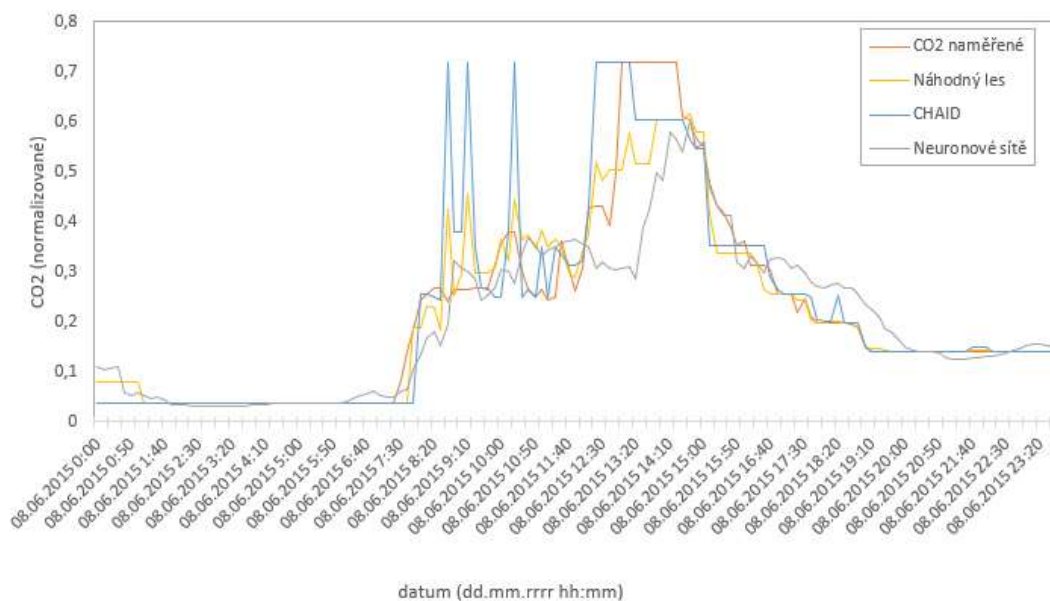
V této podkapitole jsou umístěny grafy predikovaných průběhů pro různé délky intervalu naměřených dat z provedených experimentů.

Délka jednoho dne

U grafických průběhů z obrázků Obr. 19 a Obr. 20 je viditelná roztřesenost predikovaných průběhů metodou neuronových sítí. Zajímavým místem obou grafů je několik prvních predikovaných hodnot, časový začátek grafů. Je zde vidět poměrně velká odchylka od naměřených dat, a to u všech metod.



Obr. 19 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden den 8.2.2015



Obr. 20 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden den 8.6.2015

Z hodnot kritérií uvedených v tabulkách Tab. 10 a Tab. 11 celkově nejlépe vychází metoda CHAID. Je zde vidět značné zhoršení výsledků u všech metod u datového souboru ze dne 8.6.2015. Průběh

naměřených dat v tomto souboru (Obr. 20) je oproti průběhu ze dne 8.2.2015 (Obr. 19) komplikovanější.

Tab. 10 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho dne 8.2.2015

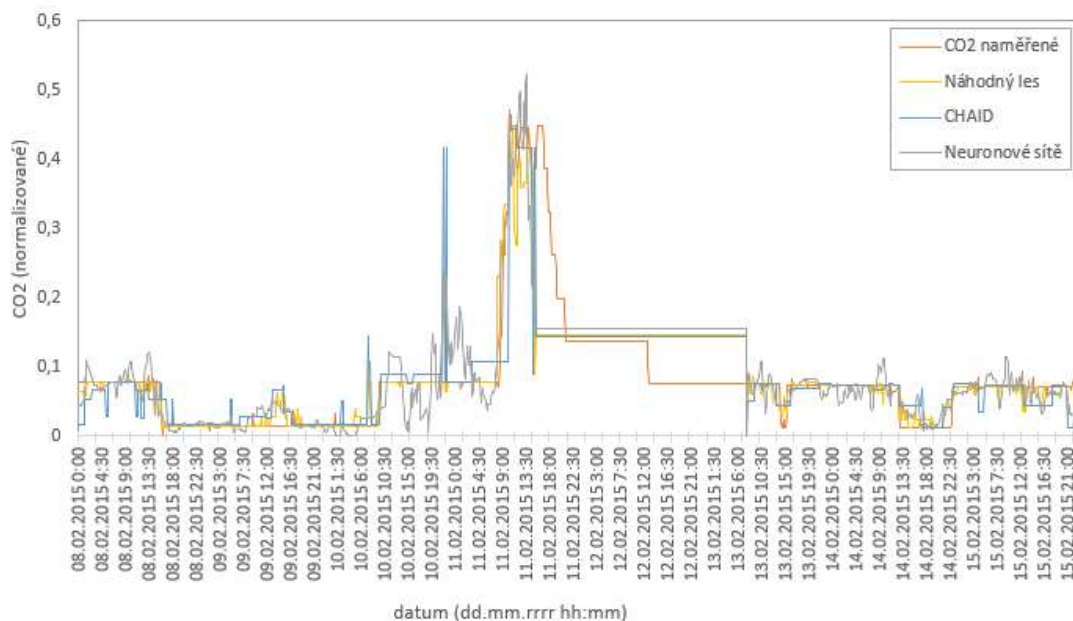
Datový soubor	LC	MAE	MSE	MAPE (%)
Náhodný les	0,966	0,002	$7,23 \cdot 10^{-5}$	5,64
CHAID	0,991	0,002	$9,75 \cdot 10^{-6}$	3,33
Neuronové sítě	0,911	0,007	$7,79 \cdot 10^{-5}$	12,33

Tab. 11 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho dne 8.6.2015

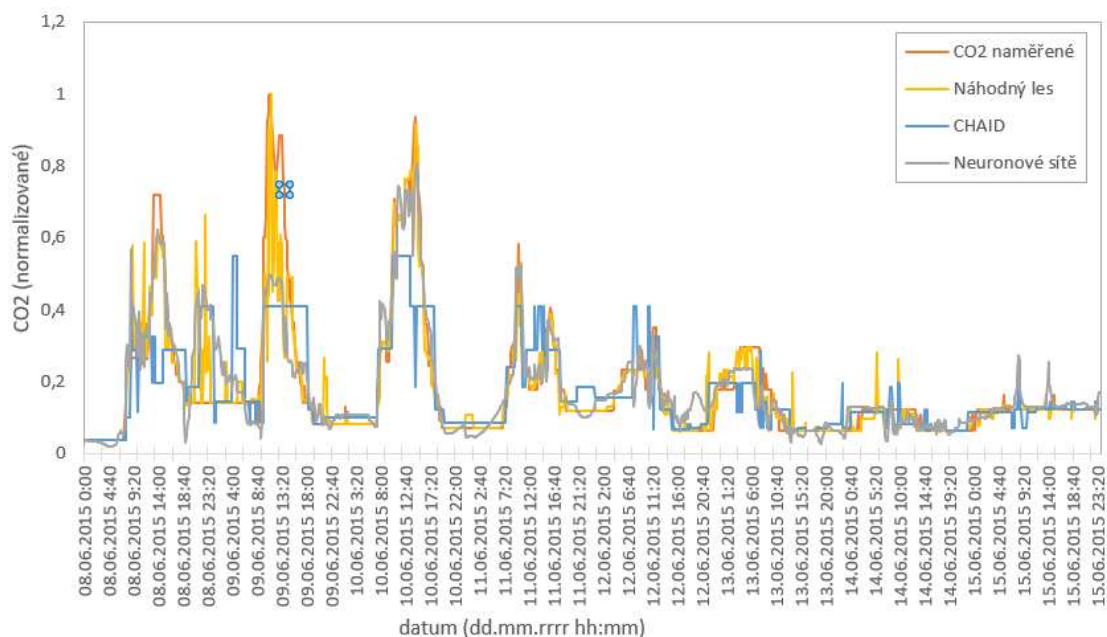
Datový soubor	LC	MAE	MSE	MAPE (%)
Náhodný les	0,905	0,045	0,068	17,71
CHAID	0,902	0,043	0,0071	12,56
Neuronové sítě	0,88	0,034	0,0087	30,52

Délka jednoho týdne

Z grafických průběhů pro časový interval jednoho týdne (Obr. 21 a Obr. 22) je mnohem více viditelná roztřesenost průběhu predikce neuronovým sítěmi. U průběhů predikce metod náhodného lesu a metody CHAID je zřejmá lineárnost celého průběhu a konstantnost průběhu i časových úsecích, kdy se konstantní i průběh naměřený.



Obr. 21 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden týden 8.2. – 15.2.2015



Obr. 22 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden týden 8.6. – 15.6.2015

Při délce intervalu dat jeden týden vychází nejlépe meto náhodného lesu (u délky intervalu dat jeden den to byla meto CHAID). Opět je zde vidět značný rozdíl ve výsledcích kritérií pro jednotlivé soubory jednoho týdne, projevila se závislost výsledku predikce na tvaru průběhu naměřených dat. Průběh hodnot CO_2 z týdne 8.6. – 15.6.2015 (Obr. 22) obsahuje mnohem více prudkých zvýšení a klesání hodnot CO_2 , než je tomu u průběhu z týdne 8.2. – 15.2.2015 (Obr. 21).

Tab. 12 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho týdne 8.6. – 15.6.2015

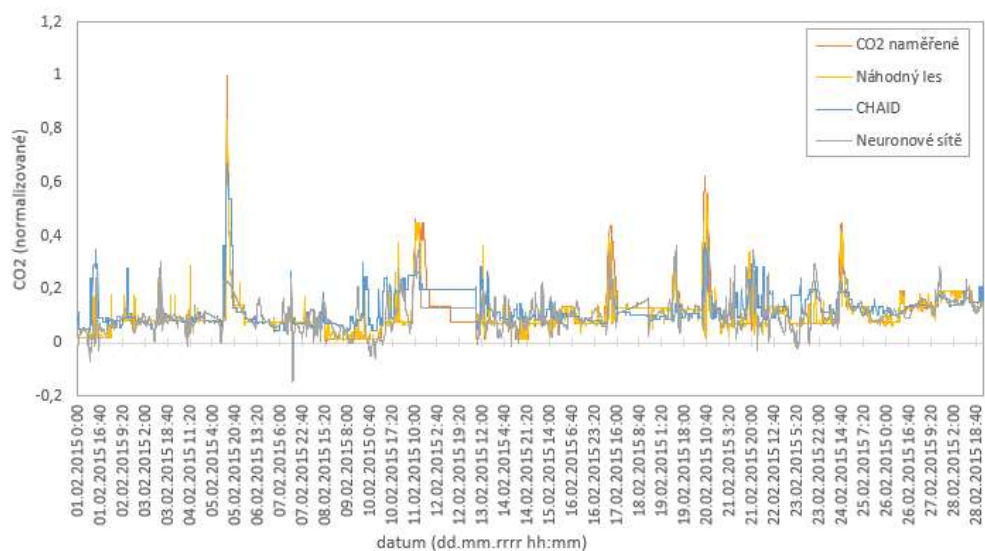
Datový soubor	LC	MAE	MSE	MAPE (%)
Náhodný les	0,892	0,035	0,0050	12,75
CHAID	0,745	0,066	0,0146	31,23
Neuronové sítě	0,852	0,057	0,0088	35,02

Tab. 13 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho týdne 8.2. – 15.2.2015

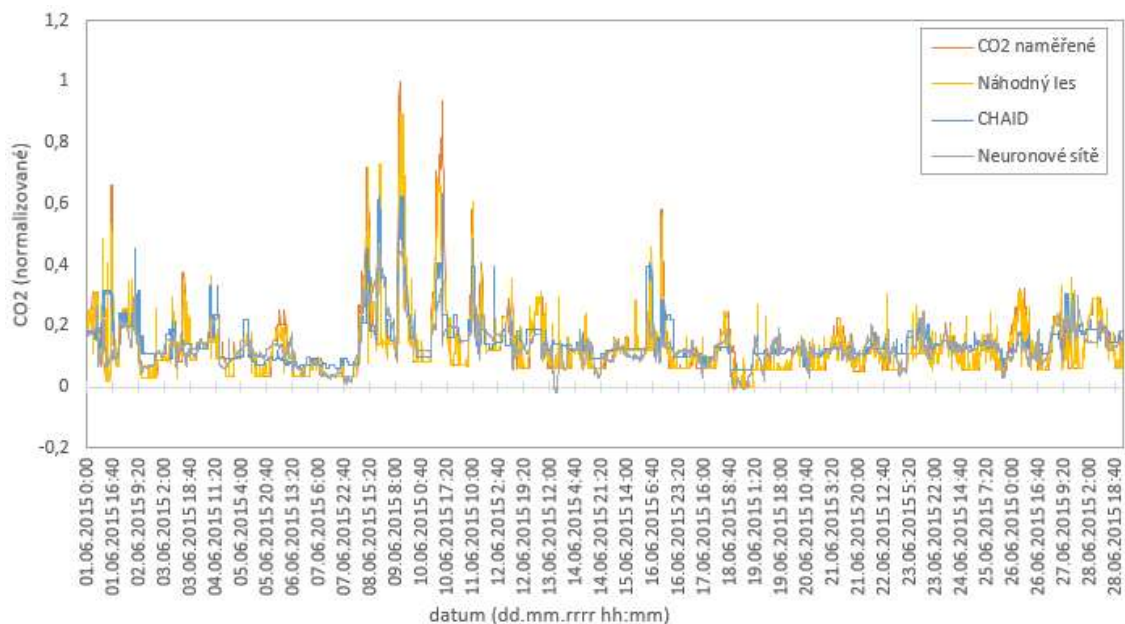
Datový soubor	LC	MAE	MSE	MAPE (%)
Náhodný les	0,867	0,021	0,0021	27,04
CHAID	0,770	0,022	0,0026	43,35
Neuronové sítě	0,734	0,029	0,0025	51,88

Délka jednoho měsíce

Z průběhu predikce koncentrace CO_2 z délky naměřených dat jednoho měsíce (Obr. 23, Obr. 24) je vidět v časech 1.2.2015 16:40, 7.2.2015 22:40 a 23.2.2015 15:20, že neuronové sítě predikovaly zápornou hodnotu koncentrace CO_2 , což je fyzikálně nemožné.



Obr. 23 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden měsíc 1.2. – 28.2.2015



Obr. 24 Grafické porovnání predikovaných průběhů metodami Náhodný les, CHAID, Neuronové sítě pro délky datového souboru jeden měsíc 1.6. – 28.6.2015

Pro tyto datové soubory nejlépe vyšla opět metoda náhodný les. Opět je zde znatelný rozdíl mezi výsledky souborů z různých měsíců.

Tab. 14 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho měsíce 1.2. – 28.2.2015

Datový soubor	LC	MAE	MSE	MAPE (%)
Náhodný les	0,723	0,022	0,0024	23,21
CHAID	0,458	0,041	0,0053	66,40
Neuronové sítě	0,530	0,044	0,0056	35,02

Tab. 15 Hodnoty kritérií použitých predikčních modelů pro datový soubor jednoho měsíce 1.6. – 28.6.2015

Datový soubor	LC	MAE	MSE	MAPE (%)
Náhodný les	0,780	0,036	0,0038	22,12
CHAID	0,533	0,061	0,0094	53,60
Neuronové sítě	0,502	0,067	0,0100	60,29

6.3 Zhodnocení experimentů

V prvním experimentu byly stanoveny vhodné parametry jednotlivých metod. Nejlépe se v tomto experimentu osvědčila metoda náhodného lesu při nastavení pěti regresních stromů (LC: 0,746; MAE: 0,020; MSE: 0,0021; MAPE: 25,22 %). S o poznáním horšími výsledky dopadla metoda CHAID a maximální hloubkou regresního stromu šesti vrstev (LC: 0,770; MAE: 0,022; MSE: 0,0026; MAPE 43,35 %). Jako poslední skončily neuronové sítě s 200 a 100 neurony ve vnitřních vrstvách (LC: 0,736; MAE: 0,028; MSE: 0,0024; MAPE 46,78 %), jejichž predikce nebyla nijak přesná. Především průběh predikovaných dat byl místy nepřesný a v celém průběhu zašuměný.

Ve druhém experimentu se ukázala závislost přesnosti predikce na délce dodaného datového souboru. Se vzrůstající velikostí datového souboru se chyba predikce zvyšuje. Všechny tři metody dosahovaly nejlepších výsledků při délce souboru jednoho dne, nejpřesnější zde byla metoda CHAID se souborem 8.2.2015 (LC: 0,991; MAE: 0,002; MSE: $9,75 \cdot 10^{-6}$; MAPE 3,33 %).

Při těchto experimentech se též ukázala poměrně silná závislost výsledku predikce na tvaru průběhu naměřených dat.

Při porovnání průběhů predikovaných dat u všech metod, oproti datům naměřeným, nedosahuje žádná z metod výrazně lepších výsledků. Nejlépe vycházejí metody náhodný les a CHAID, jejíž průběh predikovaných dat poměrně přesně kopíroval průběh dat naměřených. Tyto dva aspekty jsou důležité pro případné následné softwarové zpracování predikovaných průběhů. Nejhorší vyšla metoda neuronových sítí, jejíž průběh predikovaných dat byl v celém průběhu roztřesený a nečitelný.

Výsledky experimentů ukázaly horší výsledky metody neuronových sítí oproti metodám náhodný les a CHAID. Tento výsledek přikládám tomu, že neuronová síť nebyla naučená dostatečným objemu dat.

7 Výpočet predikce v reálném čase

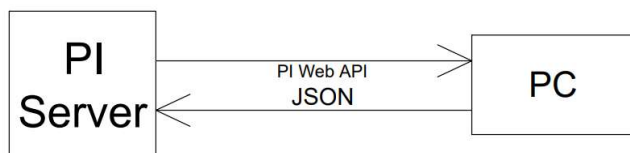
Aby bylo možné provádět výpočty predikce na vzdálených zařízeních v reálném čase, je potřeba zajistit vyčítání a zápis dat z a do PI Server taktéž v reálném čase. PI System v tomto směru nabízí více možností. Pro potřeby této práce jsem vybral PI Web API a PI Integrator for Business Analytics.

PI Web API je užitečná ve chvíli, kdy jde o komunikaci s výpočetním zařízením typu osobního počítače nebo s různými aplikacemi. Přesto je možné tuto cestu využít i s cloudovými platformami. Ty umožňují vytvoření vlastní API přímo pro vytvoření model strojového učení. To by vyžadovalo vytvoření kódu, který by zprostředkoval komunikaci mezi oběma API.

7.1 Konektivita dat, PI Web API

PI Web API má více funkcí. Může zapisovat nebo vyčítat data, ale může také vytvářet mazat PI Points, měnit atributy elementů. Pro tuto práci postačí funkce pro zapisování a čtení dat.

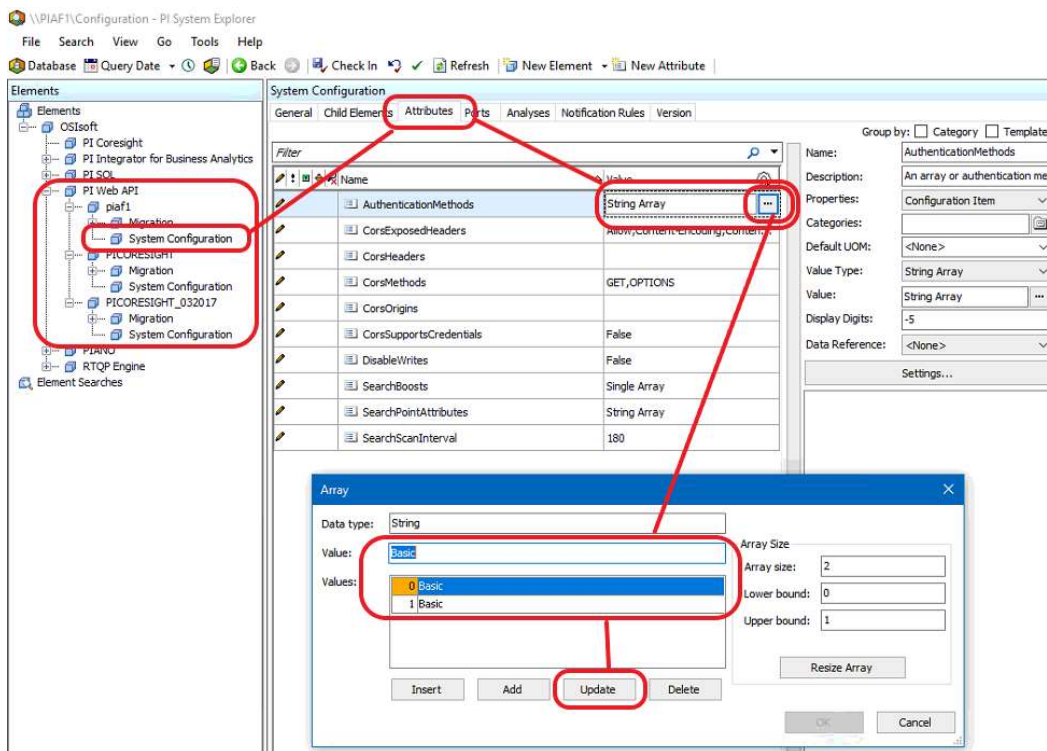
PI Web API byla tento rok nově instalována, tudíž bylo potřeba vše nastavit a zajistit oprávnění, které umožňovala použití PI Web API pro zápis a čtení dat. Bylo třeba vybrat, jaká autentifikační metoda bude použita. Prvně jsem hledal mezi metodami, které nepotřebují k autentifikaci přihlašovací údaje, jelikož jsem zpočátku žádné k dispozici neměl. PI Web API autentifikaci prostřednictvím certifikátu neumožňuje a použití ověření Anonymous bylo nepřípustné a na nastavení dalších metod jako Bearer jsem neměl oprávnění. Zbývala jediná možnost použití autentifikace Basic. Zde bylo nutné sjednat vytvoření účtu u správce PI System. Celý proces výběru autentifikační metody až po její zprovoznění trval velmi dlouho.



Obr. 25 Blokový diagram konektivity mezi PI Serverem a PC

Změna autentifikační metody

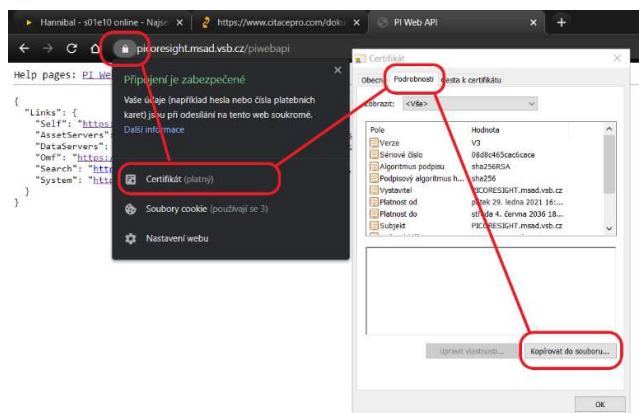
Při instalaci nástroje PI Web API je defaultně nastavena autentifikační metoda Kerberos. Pro potřeby této práce bylo nutné změnit autentifikační metodu na Basic. Změna autentifikační metody se provádí prostřednictvím nástroje PI System Explorer, databáze Configuration. Postup změny autentifikační metody je zobrazen na obrázku níže.



Obr. 26 Změna autentifikační metody na Basic

Export certifikátu

Použití certifikátu není při nastaveném zabezpečení basic nutné. Většina knihoven nebo například Postman, umožňuje použití certifikátu obejít. Cest, jak certifikát exportovat, je více. Osobně jsem použil export z webového prohlížeče, na kterém je přihlášená PI Web API. Exportovaný certifikát je typ souboru .cer. Ten lze bez dalších konverzí použít při verifikaci v jazyce Python.



Obr. 27 Postup certifikátu z webového prohlížeče

Kontrola a nastavení přístupových oprávnění

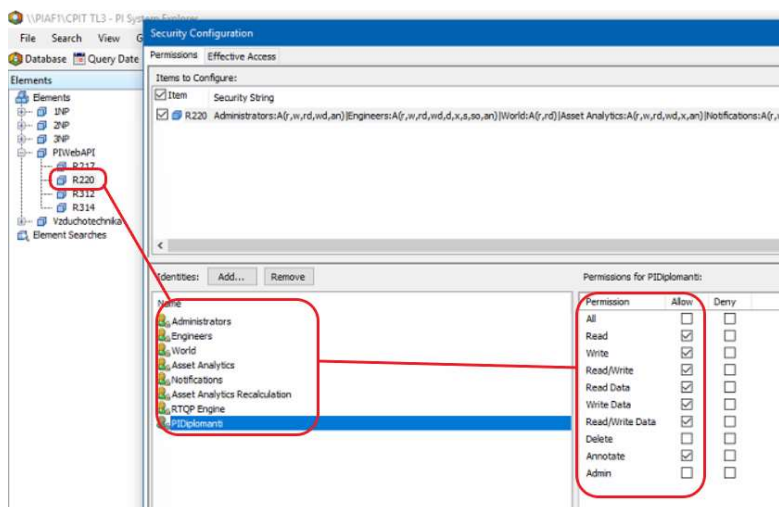
Nastavení oprávnění je třeba provést pro PI Asset Framework i PI Data Archive. Musí být povolen zápis a čtení dat jak do PI Point atributu v PI AF, tak zápis a čtení v samotném pointu v PI Data Archive. Taktéž musí být umožněno čtení a zápis jak pro uživatele (identitu), tak pro samotnou API.

Prvně jsem tedy začal nastavovat všechna oprávnění pro samotnou PI Web API. Je třeba se ujistit, že PI web API má nastavený atribut DisableWrite na False, viz. obrázek níže.



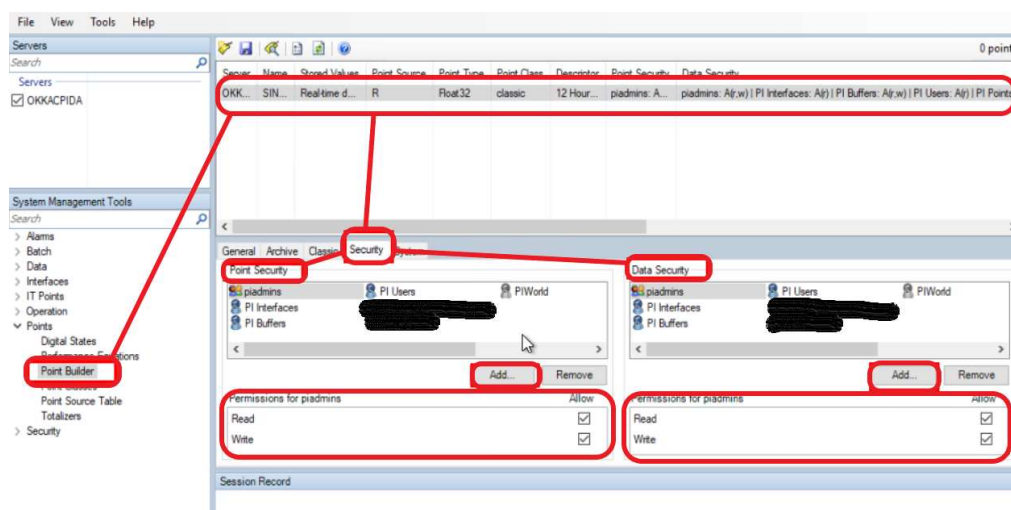
Obr. 28 Povolení zápisu pro PI Web API, nástroj PI System Explorer

Poté je nutné ověřit, zda identita nebo účet, pod kterým se budete v PI API autentifikovat, má povolené čtení a zápis do příslušných elementů v PI AF. Stačí v PI System Explorer najít příslušný element a v nastavení zabezpečení pro danou identitu oprávnění přidělit.



Obr. 29 Příklad nastavení oprávnění pro přístup k elementu v PI System Explorer

Nakonec bylo nutné nastavit oprávnění pro čtení a zápis do samotných PI Points v PI Data Archive. Pro to je použit PI SMT. V záložce *Points – Point Builder* je potřeba nejprve příslušné Points načíst. Poté se v *Security* okna *Point Security* a *Data Security* nastaví čtení a zápis pro danou identitu či uživatele do samotných PI Points.



Obr. 30 Nastavení oprávnění čtení a zápisu pro PI Points, PI SMT

Čtení dat

Ve chvíli, kdy je vše správně nastaveno a jsou vyřešeny všechny náležitosti, je možné přistoupit k samotnému vytváření requestů (požadavků). Ze začátku jsem pro vytváření prvotních requestů a seznámení se s postupy jejich vytváření používal Postman a Fiddler. Oba tyto nástroje jsem velmi rychle opustil, jelikož s nimi bylo více škody než užtku.

Kód pro komunikaci s PI Web API je psán v jazyce Python, je zde použita knihovna requests. Vytváření požadavků pro PI Web API je stanoveno dle kontrolérů, které mají definované funkce. Ty stanovují, jak mají požadavky vypadat a jaké náležitosti se v nich mají nacházet. Jejich seznam lze jednoduše najít v helpu PI Web API.



Obr. 31 Okaz pro vstup do helpu z PI Web API

Pro čtení jedné hodnoty se používá kontrol Stream, funkce GetValue. Jedná se o metodu GET, status code pro úspěšně přijatý požadavek je 200. Jeho složení je následující:

`https://picoresight.msad.vsb.cz/piwebapi/streams/{WebId}/value`

Pro výčet více hodnot najednou existuje kontrol Batch. Při čtení hodnot CO₂, relativní vlhkosti, vnitřní a venkovní teploty, stačí poslat jeden request funkce Batch místo čtyř requestů funkce GetValue. Kontrol Batch je metoda POST. Ukázka volání requestu Batch v jazyce Python.

```

url = "https://picoresight.msad.vsb.cz/piwebapi/batch"
body = {
    "CO2_220":{
        "Method":"GET",
        "Resource": "https://picoresight.msad.vsb.cz/piwebapi/streams/{WebID}/value"
    },
    "rh_220":{
        "Method":"GET",
        "Resource": "https://picoresight.msad.vsb.cz/piwebapi/streams/{WebID}/value"
    },
    "Tin_220":{
        "Method":"GET",
        "Resource": "https://picoresight.msad.vsb.cz/piwebapi/streams/{WebID}/value"
    },
    "Tout_220":{
        "Method":"GET",
        "Resource": "https://picoresight.msad.vsb.cz/piwebapi/streams/{WebID}/value"
    }
}

r= rq.post(url, auth = auth_udaje, verify = cert, json = body)

```

Obr. 32 Ukázka volání requestu Batch v jazyku python

Na místo {WebID} se uvádí WebId příslušného prvku. Pro zasílání requestů není nutné uvádět self-signed certifikát. Lze ho obejít zápisem *verify = False*.

Při začátku čtení dat jsem si všiml, že jednotlivé hodnoty nemají stejné časové známky. Nová hodnota pro danou veličinu je nejspíš uložena ve chvíli, kdy se právě naměřená hodnota významně liší od poslední uložené hodnoty.

Zápis dat

Zápis dat je proveden metodou POST. Data jsou do PI Data Archive ukládána v podobě záznamů do tagů (PI Points). Zápis je možný jak do již existujících Pointů, tak do nově vytvořených. Pointy lze vytvořit dvěma cestami:

- prostřednictvím PI System Management Tools,
- Prostřednictvím PI Web API, metoda CreatePoint.

Za účelem této práce jsem vytvořil pro ukládání hodnot predikce nové Pointy prostřednictvím PI SMT. Pointy jsou vytvořené pro místnosti R217, R220, R312 a R314. Názvy Pointů pro jednotlivé místnosti mají shodná jména, liší se pouze v čísle místnosti. Pointy vytvořené pro ukládání dat predikce koncentrace CO₂ v místnosti R220:

- CO2_predict_R220 – Point pro ukládání hodnot predikce koncentrace CO₂;
- MAE_R220 – Point pro ukládání hodnot kritéria MAE;
- MSE_R220 – Point pro ukládání hodnot kritéria MSE;
- MAPE_R220 – Point pro ukládání hodnot kritéria MAPE.

Do vytvořených Pointů se vkládají data kontrolou Stream, funkce UpdateValue. Pro úspěšné zapsání hodnoty je status code 202 nebo 204. Tento kontrol umožňuje zapisovat pouze jednu hodnotu na

jeden request. Pro zápis všech čtyř hodnot (hodnota predikce, kritéria MSE, MAE, MAPE) najednou jsem použil kontrol StreamSets, funkce UpdateValue.

Kromě samotné hodnoty je třeba zapsat i čas, ve kterém byla predikce provedena. Jak jsem popisoval v předchozí podkapitole. Jednotlivé vyčítané hodnoty mají různé časové známky. Stejně tak si lze všimnout, že čas v časových známkách je zhruba dvě hodiny zpožděný. To je způsobeno tím, že je zde použit koordinovaný světový čas – UTC (Zulu). Česká republika se nachází v časovém pásmu UTC + 2 hodiny. Proto je třeba i pro zápis hodnot do PI Data Archive použít Zulu čas. Python nabízí přímo knihovnu zulu. Při vytváření časové známky se na její konec musí připojit písmeno "Z", které značí, že se jedná o Zulu čas. Pokud je v časové známce použit standardní čas našeho časového pásma, není request přijat a k zápisu dat nedojde, šlo by o zápis dat do budoucnosti.

```
time = str(zulu.now())
timeISO8601 = time[:19]+"Z"

url = "https://picoresight.msad.vsb.cz/piwebapi/streamsets/{WebId}/value"
body = [
    {
        #CO2_pred
        "WebId": "{WebID}",
        "Value": {
            "Timestamp": timeISO8601,
            "Value": 455,
        }
    },
    {
        #MAE
        "WebId": "{WebID}",
        "Value": {
            "Timestamp": timeISO8601,
            "Value": 10,
        }
    },
    {
        #MSE
        "WebId": "{WebID}",
        "Value": {
            "Timestamp": timeISO8601,
            "Value": 100,
        }
    },
    {
        #MAPE
        "WebId": "{WebID}",
        "Value": {
            "Timestamp": timeISO8601,
            "Value": 100,
        }
    }
]
r = rq.post(url, auth = auth_udaje, verify = cert, json = body)
```

Obr. 33 Ukázka volání requestu pro zápis dat do PI Data Archive v jazyce Python

7.2 Výpočet predikce, predikční model

Místo cloudových platform, jakožto prostředku pro výpočet predikce koncentrace CO₂ v reálném čase, jsem se rozhodl použít pro výpočet prostředí jazyku Python. V tomto jazyce jsem vytvořil kód, ve kterém je zajištěn výpočet predikce, tak i komunikace s PI Web API.

Načtení a příprava datového souboru

Jako historická data jsou zde použity datové soubory, které byly naměřeny v období od 25.6.2018 do 29.6.2018 na třech různých místnostech R104, R203, R204 na budově dřevodomku. Jedná se o data v surovém stavu, která nejsou nijak předzpracována. Perioda měřených dat je jedna minuta, celý soubor čítá 7200 řádků.

Načítání datového souboru je provedeno z formátu CSV. Jak pro načtení souboru, tak pro práci s poli, je v celém kódu použita knihovna numpy. Načtený datový soubor je rozdělen do dvou polí – prediktory a cíl predikce. Dále je soubor prostřednictvím funkce `train_test_split` rozdělen na testovací a trénovací data. Použil jsem zde 80 % trénovací data, 20 % testovací. Na trénovacích datových souborech je provedeno naučení modelu. Následně se naučený model otestuje na testovacích datech. Způsoby rozdělení souboru na trénovací a testovací části mohou být různé.

Před samotným učením modelu je proveden další krok předzpracování datového souboru - normalizace dat. Normalizace obvykle přináší lepší výsledky predikce. Některé predikční modely toto předzpracování vyžadují (ANN). Při normalizaci jsou hodnoty prediktoru či cíle převedeny do sjednoceného intervalu. Ten obvykle bývá nastaven na hodnoty 0 až 1 nebo -1 až 1. Důvod normalizace dat je následující - některá data použitá v predikci mohou mít hodnoty v řádech desítek tisíců, jiná pouze v desítkách nebo jednotkách. To je velký rozdíl ve velikostech. Normalizací je rozptýl hodnot pro všechna data sjednocený [53].

Dále by se mohlo použít nalezení chybějících hodnot nebo nějaké úpravy z hlediska sloupců. Datový soubor jsem si vizuálně prohlédl a zformátoval do požadovaného stavu v Excelu. Takže další předzpracování nebylo nutné.

Učení a testování modelu strojového učení

Postup je následující – naučení modelu na trénovacích datech a následně jeho testování na testovacích datech. Na výsledky testovací množiny jsou použity výpočty kritérií, čímž je model ohodnocený.

Pro predikci dat koncentrace CO₂ dle výsledků experimentů provedených v kapitole 6 a zkušeností získaných při experimentech s cloudovými nástroji strojového učení, volím predikční model náhodného lesu. Parametry: počet regresních stromů 5, maximální hloubka stromu 10.

Tab. 16 Výsledky kritérií predikčního modelu náhodný les v jazyce Python

Datový soubor	LC	MAE	MSE	MAPE
R104	0,979	0,011	0,002	2,03
R203	0.989	0,0062	$6 \cdot 10^{-3}$	3,12
R204	0,988	0,0067	$6 \cdot 10^{-3}$	1,96

Při použití této metodiky dochází k zásadnímu problému. Jakmile je naučený model nasazen a je na něm zahájen výpočet predikce v reálném čase, jeho přesnost dramaticky klesne. Nepřesnost predikce byla při testování modelu mezi 2 – 3 %. To však bylo testování na části datového souboru, na kterém byl samotný model naučen. Při zahájení výpočtů predikce z dat ze CPIT TL3 klesla přesnost predikce na slabých 20 %. Problém se skrývá v učícím datovém souboru, přesněji v rozsahu jeho hodnot. Datový soubor, na kterém byl model naučen, má určitý rozsah jak naměřených hodnot CO₂, tak i hodnot prediktorů. V případě datového souboru použitého pro trénování je rozsah hodnot CO₂ 539,8 ppm až 605,6 ppm. Avšak data, která byla v tu dobu měřena na CPIT, měla hodnoty CO₂ mezi 400 - 500 ppm. Model v tu chvíli tedy predikoval, ale jeho výsledky byly omezeny na rozsah CO₂, na kterém byl naučený. Výsledky predikce byly nedostačující (MAE: 83,3; MSE: 6937,7; MAPE: 16,2 %). Výsledky nijak nezlepší ani normalizace dat. V předzpracování jsou všechna data normalizována do intervalu 0 až 1. Avšak hodnoty, které byly měřeny na CPIT TL3, měly po normalizaci hodnoty někdy i záporné nebo přesahující hodnotu 1. Řešení problému s predikcí v reálném čase by vyžadovalo implementaci adaptivní normalizace [54].

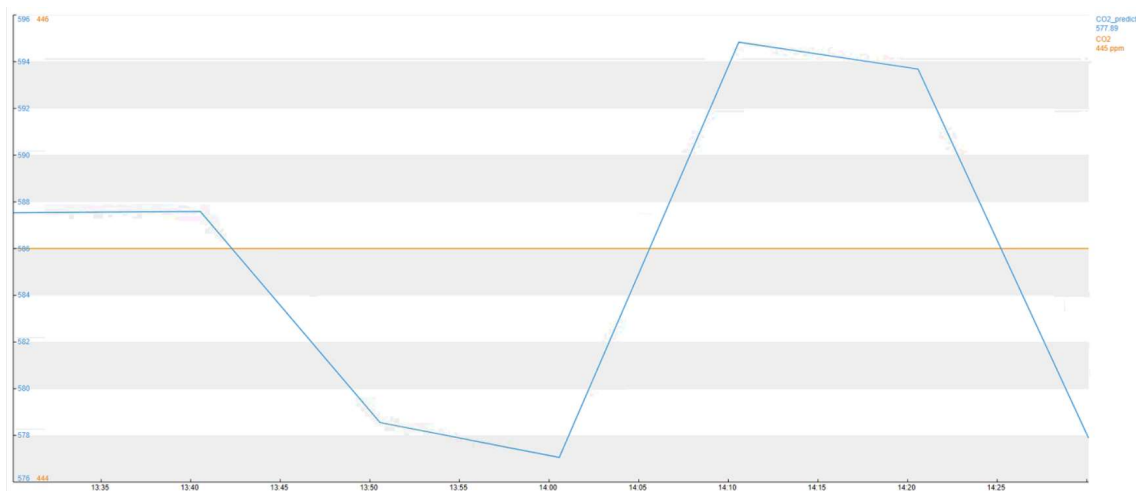
Dramatický pokles přesnosti predikce demonstruji v následujícím experimentu. Učení modelu provedu na celém datovém souboru, který čítá 7200 údajů, rozsah hodnot CO₂ je 551,5 ppm až 670,4 ppm (datový soubor R203). Poté provedu predikci s konkrétními daty (z datových souborů R104 a R204), u kterých je koncentrace CO₂ vyšší nebo nižší, než je rozsah CO₂ trénovacího souboru. Zároveň provedu výpočet predikce na datech ze souboru R104, která mají hodnoty CO₂ v rozsahu hodnot CO₂ trénovacího souboru. Data jsou předzpracována prostřednictvím normalizace v intervalu 0 až 1.

Tab. 17 Porovná přesnosti predikce pro různé rozsahy hodnot CO₂

-----	LC	MAE	MSE	MAPE [%]
CO ₂ nižší než 551,5ppm	-0,34	0,26	0,07	62
CO ₂ vyšší než 670,4 ppm	0,28	0,39	0,17	48
CO ₂ v rozsahu 551,5 ppm až 670,4 ppm	0,78	0,17	0,037	14

Výsledky toho experimentu ukazují, jak se liší výsledky predikce, pokud jsou naučenému modelu předložena data z jiných datových souborů, než na jakých byl model naučen. Je zde vidět, že přesnost predikce prudce poklesla, pokud byl rozsah hodnot CO₂ testovacího souboru mimo rozsah hodnot množiny tréninkových dat. Hodnoty predikce CO₂ nepřesáhly hranici 650 ppm, nebyly nižší než 558,8 ppm.

Na Obr. 32 je vidět rozdílnost mezi predikovaným a naměřeným průběhem hodnot koncentrace CO₂ v reálném čase v časovém horizontu jedné hodiny.



Obr. 34 Graf porovnání průběhů predikovaných a naměřených hodnot v reálném čase, PI Vision

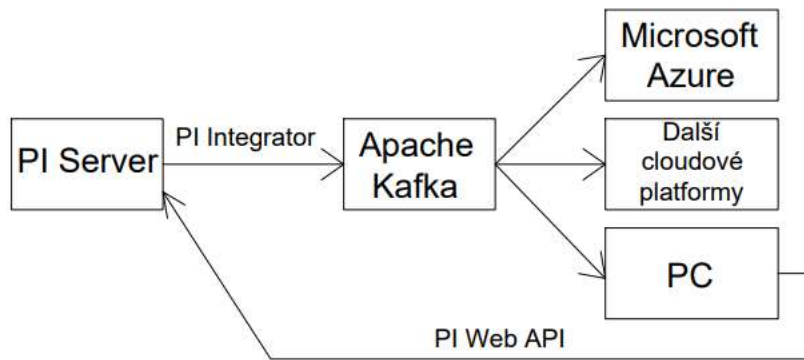
Čas, který uběhne za dobu od odeslání požadavku pro čtení dat až po zapsání výsledků predikce zpět do PI Server, není možné přesně změřit. Čtení dat prostřednictvím PI Web API je řízeno dle časové periody. Toto spouštění nemá žádnou zpětnou vazbu o tom, kdy nebyla hodnota určité veličiny v PI System přepsána na novou hodnotu. Navíc časové známky pro všechny čtyři čtené hodnoty jsou různé. Při zapisování hodnot predikce do PI Server se používá časová známka s časem, který byl aktuální ve chvíli vytváření requestu pro zápis dat. Z těchto důvodů neexistuje žádný přesný způsob, jakým by šlo celý proces přesně změřit. Je tu však možnost, jak celý proces změřit manuálně. Provedl jsem řadu ručně spuštěných výpočtů predikce. U těch jsem sledoval čas, který uběhne od odeslání požadavku pro čtení dat až po chvíli, kdy se nová hodnota objeví ve vizualizaci. Po deseti pokusech je průměrný čas 4,5 sekundy.

7.3 PI Integrator for Business Analytics, Apache Kafka

Samotný PI Integrator může přenášet PI data do různých cílových uložišť nebo služeb. Cílové objekty musí být předem nastaveny a nakonfigurovány, jsou označeny jako Targets. Jejich seznam a podrobnější popis PI Integrator je uveden v kapitole 3.4.. Data lze sdílet přímo s uložišti cloudových platforem. Pro tuto práci je nejzajímavější sdílení dat do Microsoft Azure a Apache Kafka.

Kombinací PI Integrator a Apache Kafka lze vytvořit streamování PI dat, které má oproti PI Web API řadu výhod. Hlavní přednost je v tom, že data z Apache Kafka mohou být poskytována uživatelům, kteří nemají v PI System žádné oprávnění a nemají žádnou znalost tohoto systému. Apache Kafka zde působí v roli producenta, ten poskytuje data koncovým uživatelům – konzumentům. PI Integrator nedokáže zapisovat data nebo vytvářet Pointy. Proto je pro zpětný zápis predikovaných hodnot použita PI Web API.

Tento postup je inspirován publikací vydanou společností OSIsoft [55].



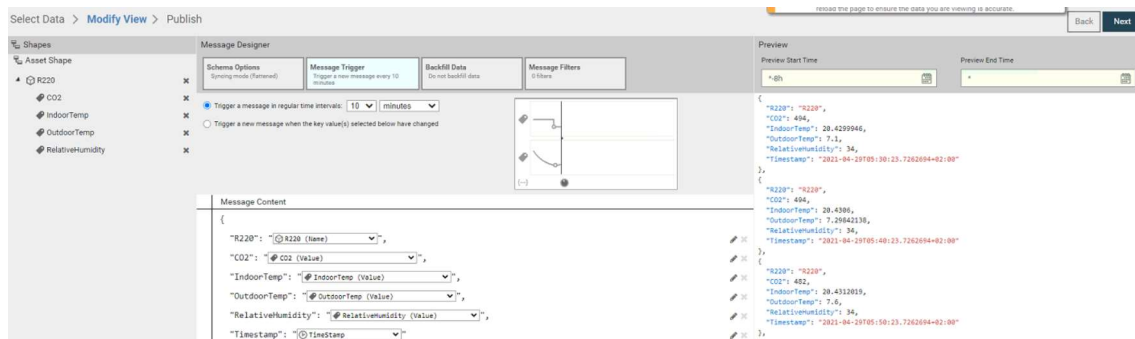
Obr. 35 Blokový diagram komunikace mezi PI Server a koncovým uživatelem.

Aby mohla být data poskytnuta, je nutno vytvořit tzv. View. To má tři podoby:

- Asset View,
- Streaming View,
- Event View.

V rámci zajištění predikce jsou zajímavé Asset View a Streaming View. Prostřednictvím Asset View je vytvořen ucelený soubor historických dat. Ta je možné použít pro analýzu a natrénování modelu strojového učení. Streaming View poté poskytuje data, ze kterých je počítána predikce.

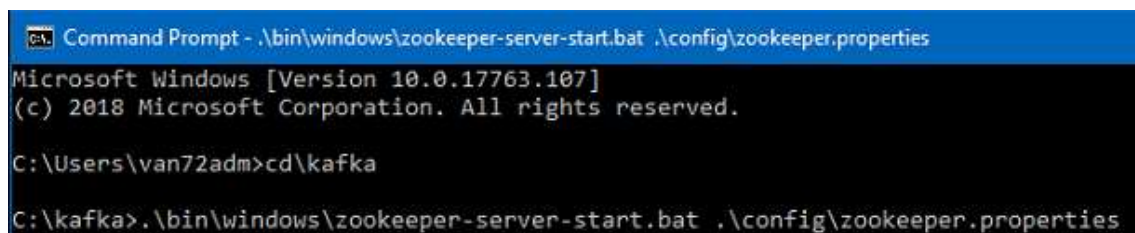
U Streaming View lze nastavit způsob, jak má být streamování spouštěno. Na výběr jsou dvě možnosti: streamování dat v nastaveném časovém intervalu nebo při změně jedné z vybraných hodnot. View, které jsem vytvořil pro demonstraci funkce streamování dat na Apache Kafka, je nastaveno na odesílání dat v intervalu 10 minut a je dostupné pod názvem CPITTL3_R220_predict.



Obr. 36 Nastavení Trigger pro Streaming View, PI Integrator for Business Analytics

Spuštění Apache Kafka

Na serveru 158.196.109.334 je vytvořen server Apache Kafka a Zookeeper. Spuštění serveru se provádí prostřednictvím příkazové řádky a spuštění příkazů ze složky *kafka*, která je umístěna na disku C.

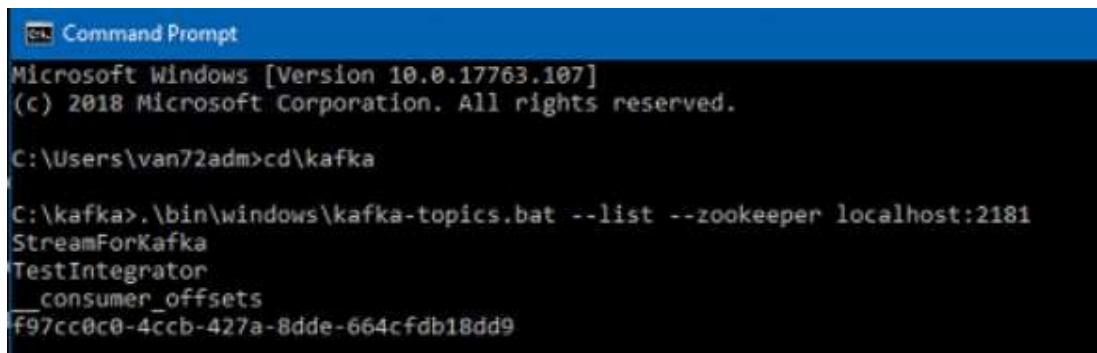


Obr. 37 Ukázka spuštění serveru Zookeeper z příkazové řádky

Postup spuštění Apache Kafka se skládá ze čtyř kroků:

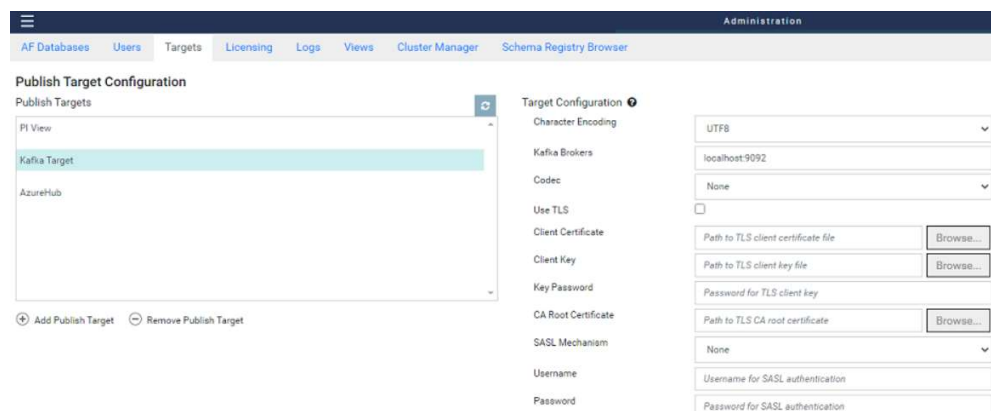
- Spuštění serveru Zookeeper
- Spuštění serveru Apache Kafka
- Spuštění konzumenta

Je možný ještě čtvrtý krok, ve kterém je vytvořen Topic. Ten zde neuvádím, jelikož jsem Topic už vytvořil. Je dostupný pod názvem StreamForKafka.



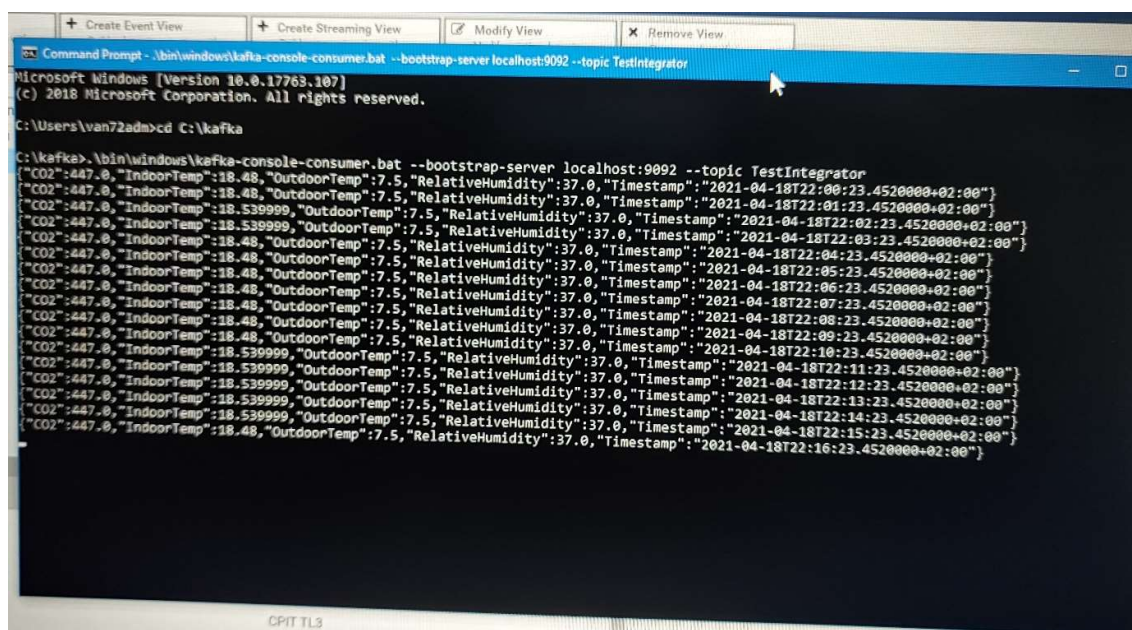
Obr. 38 Seznam vytvořených Topics

Dále je v PI Integrator vytvořen Target pro spuštění server Apache Kafka. Před zahájením streamování je dobré tento target verifikovat. To je provádí v okně na Obr. 36 odkazem "Verify Kafka Writer".



Obr. 39 Vytvořený Target v PI Integrator pro Apache Kafka

Ve chvíli, kdy je spuštěný Stream View v PI Integrator a Apache Kafka je nastavený a spuštěný, stačí spustit konzumenta v příkazové řádce. Tím je zahájeno čtení streamovaných dat. Na Obr. 40 je okno příkazové řádky se spuštěným konzumentem a výpisem streamovaných dat. Jde o hodnoty koncentrace CO₂, vnitřní teploty, venkovní teploty, relativní vlhkosti a přiřazená časová známka.



Obr. 40 Výpis streamovaných dat z PI Integrator a Apache Kafka do okna příkazové řádky

Jednotlivé příkazy pro spuštění serverů Zookeeper, Apache Kafka a spuštění konzumenta:

- Zookeeper server - .\\bin\\windows\\zookeeper-server-start.bat .\\config\\zookeeper.properties
- Kafka server - .\\bin\\windows\\kafka-server-start.bat .\\config\\server.properties
- Consumer - .\\bin\\windows\\kafka-console-comsumer.bat --bootstrap-server localhost:9092 --topic TestTopic

8 Vizualizace

Vizualizace, vytvořená v rámci této práce, se zabývá dvěma byty, nacházejícími se na budově CPIT TL3. CPIT TL3 je nově vystavěná budova, která má doplnit potřeby pro výuku dosavadní fakulty FEI. Je vybavena sofistikovaným building management systémem, fotovoltaickou elektrárnou, rychlodobíjecími stanicemi pro elektrická vozidla, možnostmi sledování a řízení energetických toků a integrovanou rozsáhlou sítí rozmanitých technologií senzorických systémů. Systém budovy též umožňuje vysoko objemové datové přenosy a ukládání velkých dat [52].

Vybavení učeben je zaměřeno na tři části:

- Automotive – laboratoře, se zaměřením na e-mobilitu
- Smart Factory – 4.0, IoT, digitální dvojče, zpracování velkých dat, robotika, prediktivní údržba a další
- Home Care – pozorovací byty pro sledování behaviorálních projevů osob



Obr. 41 Budova CPIT TL3

8.1 Popis vizualizace

Tato vizualizace je zaměřena na byty, nacházející se na budově CPIT TL3 ve druhém a třetím patře. Byty slouží jak pro výuku SmartHome technologií, tak pro výuku a vývoj technologií SmartHomeCare. Zde uvedená vizualizace je vytvořena v nástroji PI Vision.

Samotný byt je tvořen čtyřmi místnostmi:

- vstupní hala,
- koupelna,
- kuchyně spojená s obývacím pokojem,
- ložnice.

K bytům jsou spjaty i další místnosti. První dvě místnosti se nacházejí ve druhém patře. Jednou z nich je provozní místnost, ve které jsou umístěny rozvaděče. Druhou místností je chodba, která spojuje vstup do bytu a provozní místnost. Dále se za stěnami bytů nacházejí chodby, skrze které lze upravovat instalaci SmartHome systémů a umisťovat do bytů další sensory a jiná zařízení. Ve vizualizaci jsou obsaženy i obrazovky pro vzduchotechnická zařízení bytů.

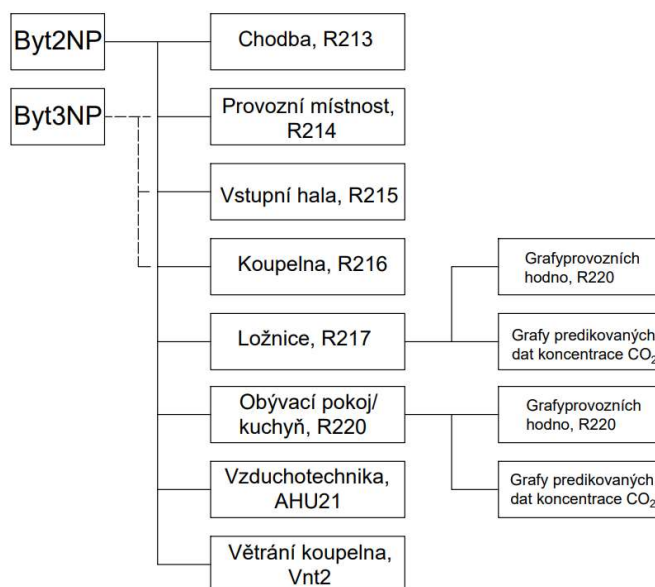
Značení místností bytu na druhém patře:

- Místnost 213 [R213] – chodba propojující byt a provozní místnost
- Místnost 214 [R214] – provozní místnost
- Místnost 215 [R215] – vstupní hala
- Místnost 216 [R216] – koupelna
- Místnost 217 [R217] – ložnice
- Místnosti 218, 219 [R218] [R219] – chodby pro úpravu/instalaci SmartHome systémů
- Místnost 220 [R220] – kuchyně s obývacím pokojem
- Vzduchotechnika [AHU21]
- Větrání koupelny [Vnt2]

Značení místností bytu na třetím patře:

- Místnost 311 [R311] – vstupní hala
- Místnost 312 [R312] – kuchyně s obývacím pokojem
- Místnost 313 [R213] – koupelna
- Místnost 314 [R314] – ložnice
- Místnosti 315, 316 [R315], [R316] – chodby pro úpravu/instalaci SmartHome systémů
- Vzduchotechnika [AHU31]
- Větrání koupelny [Vnt3]

Rozdělení vizualizace zobrazuje blokové schéma:



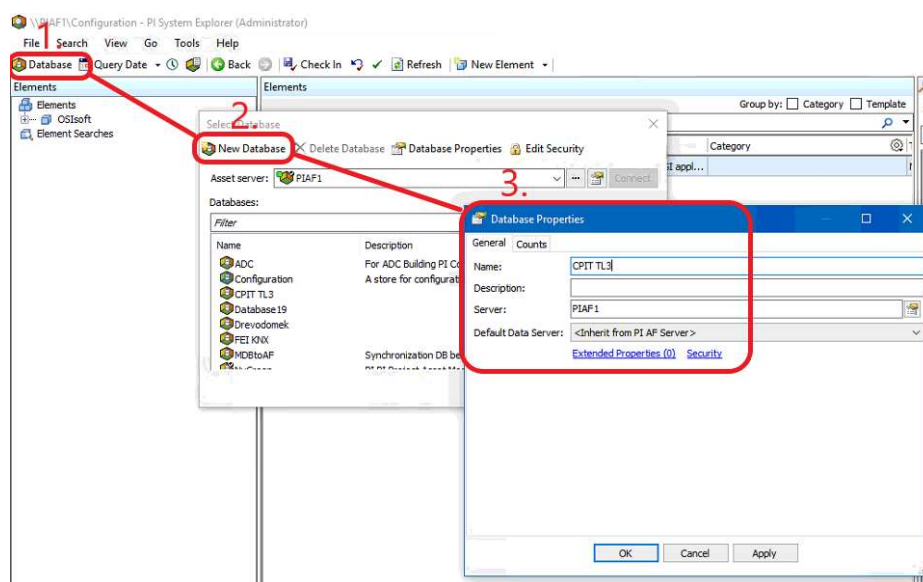
Obr. 42 Blokové schéma rozvržení vizualizace

Rozložení vizualizace je pro oba byty stejné. Vizualizace je zaměřena především na místnosti ložnice (R217, R314) a obývací pokoj s kuchyní (R220, R312), jelikož v těchto místnostech je instalována většina sensorů a detektorů. Taktéž je na tyto dvě místnosti zamýšlena predikce hodnot koncentrace oxidu uhličitého.

8.2 Vytvoření databáze pro vizualizaci

Po přesunu dat ze systému BACnet do PI System jsou tato data nepřehledná. Nalezení požadovaného tagu trvá poměrně dlouho, jelikož jejich pojmenování je souhrn zkratk, vytvořených při zavádění instalace systémů Smart Home na této budově. Navíc PI Vision vyhledávání tagů neumožňuje, tudíž je nutné vytvořit přehlednou a strukturovanou databázi, do které budou jednotlivé tagy vloženy v podobě atributů.

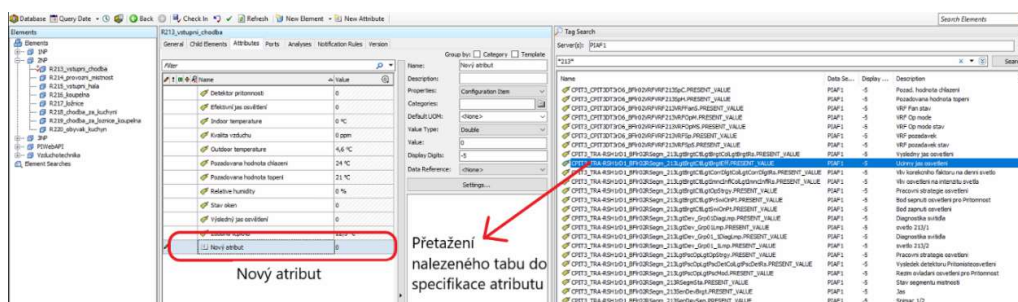
Novou databázi je možno vytvořit v programu PI System Explorer, kde se v horní liště nachází záložka *Database*. V okně *Select Database* je záložka *New Database*. Po zadání jména nové databáze a serveru, ke kterému databáze přísluší, je možné vytvořit novou databázi. Postup je znázorněn v obrázku níže.



Obr. 43 Postup vytvoření nové databáze

Do nově vytvořené databáze se vloží '*Element*' a '*Child Element*', čímž se vytvoří stromová struktura databáze. Do každého elementu je možné vložit v záložce '*Attributes*' jednotlivé tagy (vytvořit atributy elementu). Do každého elementu je možné vložit libovolný počet atributů. Postup vytvoření atributů je zobrazen na obrázku níže.

Rozvržení databáze CPIT TL3 je jednoduché. Každé patro má svůj element, v těchto elementech jsou child elementy pro jednotlivé místnosti. Samostatně jsou umístěny elementy Vzduchotechnika a PI Web API.



Obr. 44 Postup vytvoření nového atributu prvku databáze

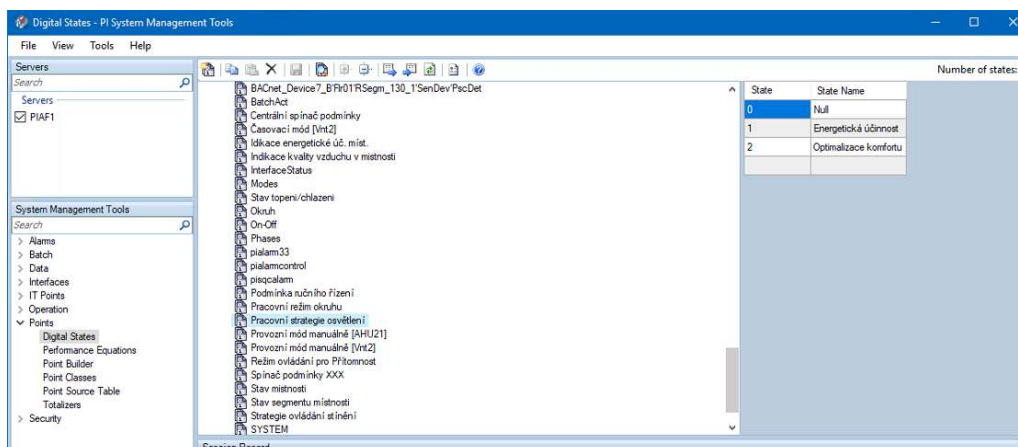
8.3 Konverze datových typů

Data reprezentující provozní stavy jsou po přesunu do PI System v podobě čísel datového formátu Int32. Samotné číslo je pro uživatele nic nevypovídající, definuje však samotný provozní stav. Pro změnu číslíkové reprezentace na textovou je nutné daný Integer převést z formátu Integer na formát Digital a přiřadit mu tabulku pro překlad číslíka na text (Digital States). Postup převodu sestává z několika kroků:

- Nalezení původní tabulky Multiplestates, přiřadit k číslům stavy.
- Vytvoření tabulky Digital States v PI System Management Tools.
- Konverze datového typu Int32 na Digital a přiřazení příslušné tabulky.

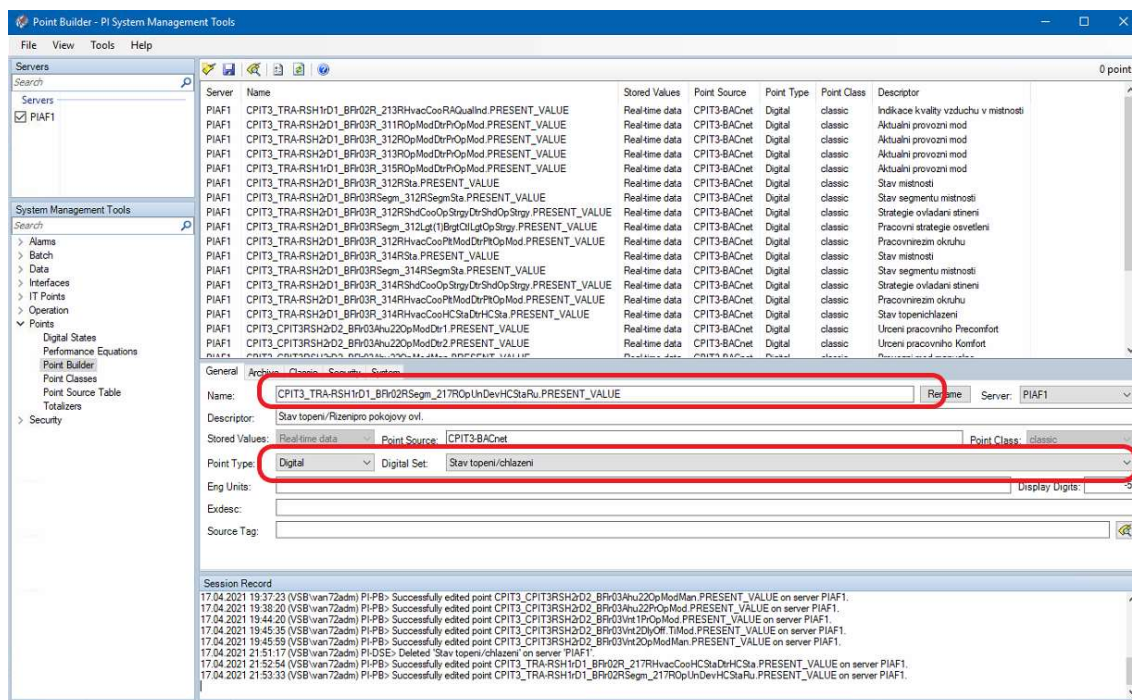
Legendu ke stavům jsem hledal v programu Inneasoftware BACnet Explorer. Ten obsahuje kompletní seznam všech hodnot, které se v systému BACnet nachází. Nevýhodou je, že tento program nenabízí žádný vyhledávač, takže hledání v něm je časově dosti náročné. Po nalezení žádaného stavu se tabulka nachází v atributu "State text". Seznam tabulek, použitých v této práci, je uveden v Příloha 1.

Vytváření Digital States tabulek a konverze datových typů se provádí v nástroji PI System Management Tools. V záložce *Points* se nachází podzáložka *Digital States*, zde se editují a vytvářejí tabulky Digital States. Je potřeba zde zmínit, že provozní stavy datového typu Int32 jsou indexovány od jedničky. Tabulky Digital States jsou však indexovány od nuly. Proto jsem v tabulkách pro index nula psal hodnotu Null.



Obr. 45 Vytvoření tabulky Digital States

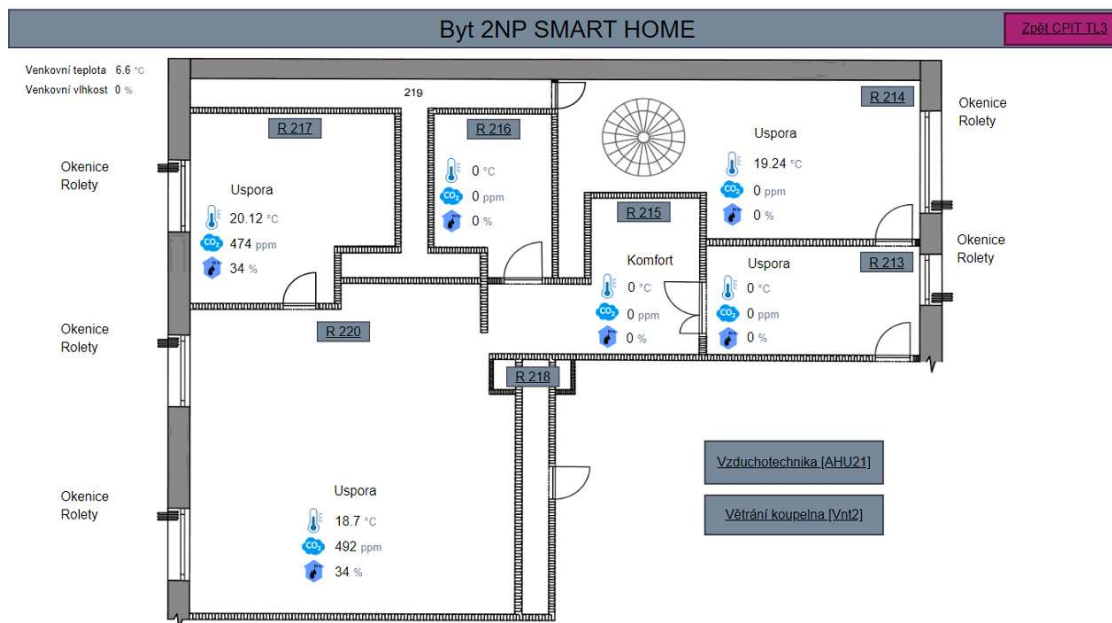
Po vytvoření tabulky se v podzáložce *Point Builder* vyhledá příslušný Point (zadá se zde jeho celý název), změní se jeho datový typ na Digital a je mu přiřazena příslušná tabulka. Po uložení je point přetypován a jeho prezentační hodnota je v textové podobě.



Obr. 46 Přetypování Pointu

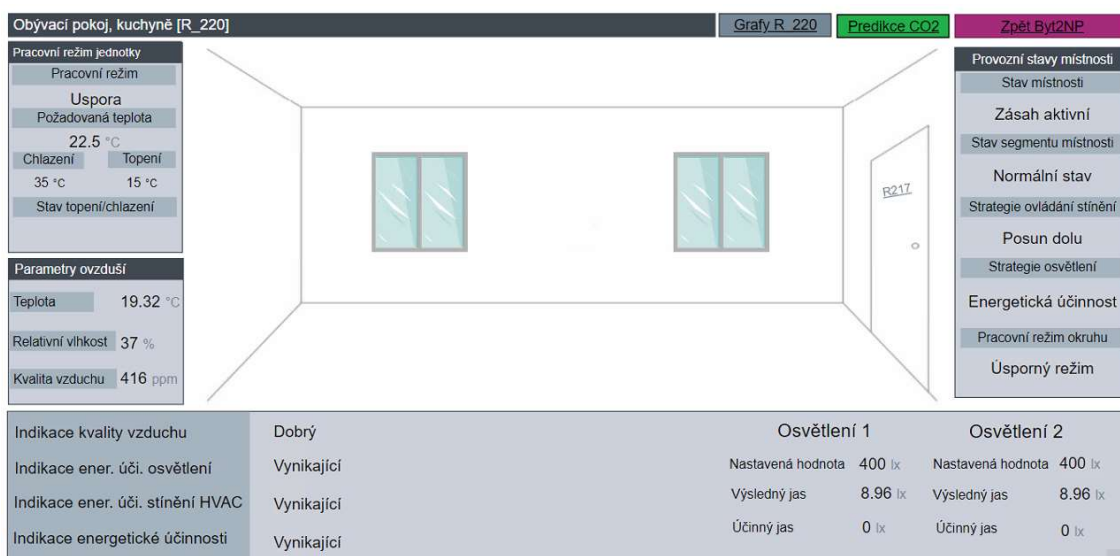
8.4 Vytvořená vizualizace

Úvodním oknem vizualizace je obrazovka s výběrem bytů (při rozšiřování vizualizace zde budou uvedeny i další místnosti či patra). Po vstupu do určitého bytu je zobrazena obrazovka s půdorysem bytu. Tento půdorys obsahuje kromě provozních hodnot i odkazy pro vstup do jednotlivých místností.



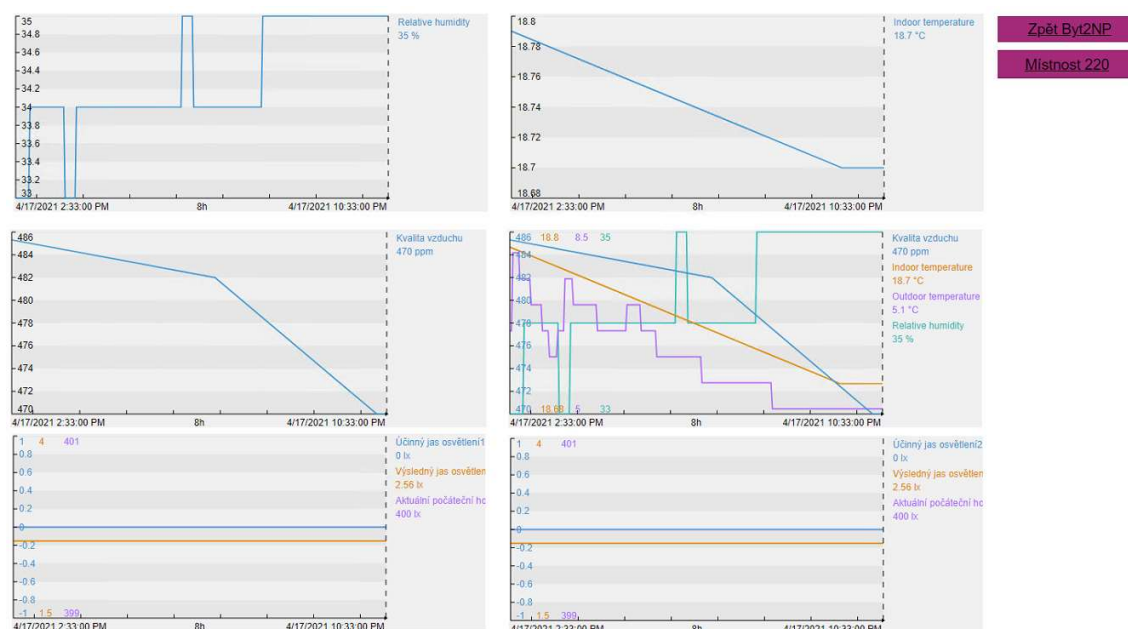
Obr. 47 Úvodní obrazovka bytu na druhém patře

V jednotlivých oknech jsou přehledně zobrazeny všechny parametry provozních funkcí týkající se dané místnosti. Ve středu obrazovek se nacházejí náčrty místností z pohledu uživatele stojícího v místnosti. V horní liště jsou umístěny tři tlačítka, sloužící jako odkazy pro přesun do jiných oken (grafy provozních hodnot dané místnosti, predikce hodnot koncentrace CO₂, návrat na úvodní stránku bytu).



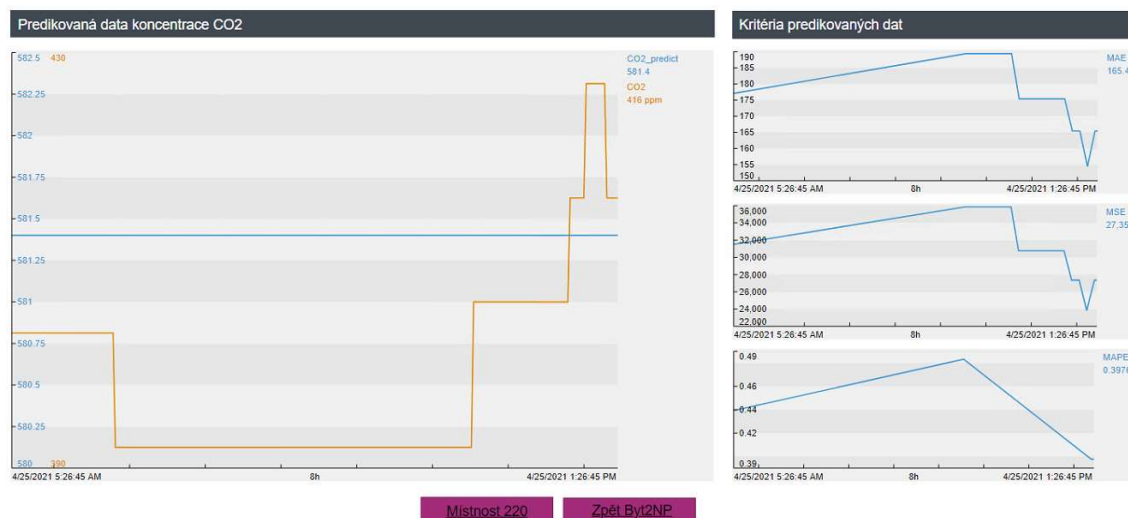
Obr. 48 Místnost 220, obývací pokoj s kuchyní

K místnostem obývací pokoj a ložnice je navíc přiřazena další obrazovka zobrazující grafy provozních hodnot. Vstupuje se do ní skrze odkaz na tlačítku "Grafy_R_220". Obrazovka obsahuje tlačítka s odkazy pro zpětný návrat do úvodního okna místnosti nebo bytu.



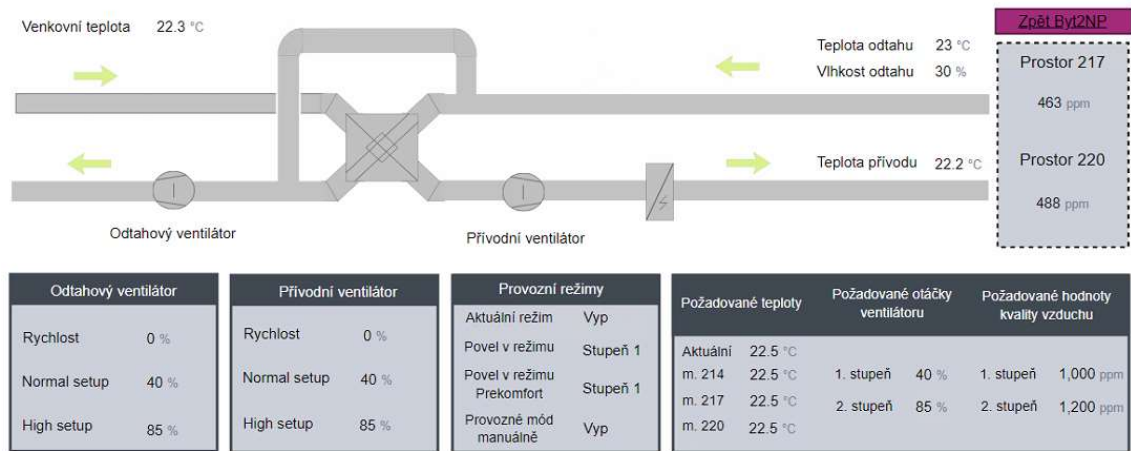
Obr. 49 Grafy pro místnost 220

Dále je k těmto místnostem přiřazena obrazovka obsahující grafy, které představují predikované hodnoty koncentrace CO_2 a také kritéria této predikce. Obrazovka obsahuje tlačítka s odkazy pro zpětný návrat do úvodního okna místnosti nebo bytu.



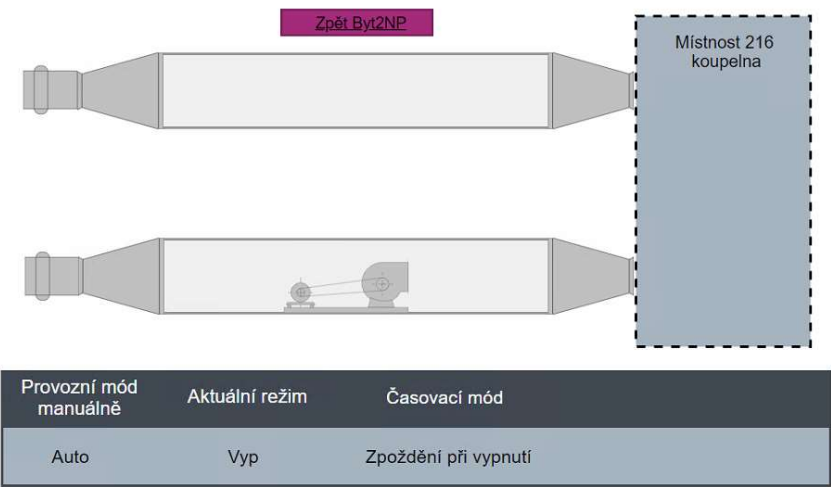
Obr. 50 Grafy predikce koncentrace CO_2 a příslušných kritérií

V obrazovce pro vzduchotechnické zařízení jsou zobrazeny hodnoty provozních stavů a nastavené parametry vzduchotechnického zařízení pro byt na daném patře.



Obr. 51 Obrazovka vzduchotechnického zařízení na druhém patře

Obrazovka pro větrání z místnosti koupelna (R216) zobrazuje pouze nastavený provozní mód a aktuální režim a časový mód.



Obr. 52 Obrazovka provozně technických funkcí koupelny na druhém patře

Problémy při vytváření vizualizace

Při vytváření vizualizace jsem narazil na několik problémů. Prvním byl fakt, že několik hodnot nebylo možné dohledat, a to ani v DesigoCC (program s vizualizací a ovládáním celé budovy CPI TL3). Jde o hodnoty reprezentující pozici rolet a okenic, hodnoty detektorů přítomnosti a stav otevřenosti oken. Osobně jsem tyto hodnoty dlouhou dobu hledal. Nakonec jsem vše konzultoval se správcem budovy. Ani při společné snaze se nám nepodařilo data dohledat. Jednou z teorií je to, že poloha rolet a okenic není statická veličina. Při aktivaci okenic nebo rolet přes pokojové tlačítko jsou uvedeny do pohybu přednastavenou hodnotou akčního zásahu (pohyb se zastaví přes pokojový ovladač nebo koncovým spínačem, neurčuje se jich výška ani poloha). Pokud je v DesigoCC přesně nastavena pozice (např. 35

%), je vypočtena hodnota akčního zásahu, který vykoná pohyb z nulové pozice (zcela vytažené), ale samotná hodnota akčního zásahu není nikde zaznamenána. Zajištění těchto hodnot by nejspíš možné bylo, byla by však nutná účast odborníka ze Siemens, jenž by vše nastavil.

Dále není možné dohledat data z tlačítek KNX, používaná v místnostech pro ovládání osvětlení, okenic a rolet. KNX instalace je totiž s BACnet propojena skrz sběrníkový modul, který tyto hodnoty nepropouští. Navíc nejsou jednotlivé komponenty KNX propojeny standardní sběrníky KNX, ale sběrníky společnosti Siemens.

Dalším problémem jsou časté výpadky dat ve vizualizaci. Místo hodnot je zobrazen nápis "No Data". Nějakou dobu jsem tuto poruchu sledoval. Nejčastěji se projevila, pokud došlo ke změně v databázi, nejčastěji při změně jména atributu (tagu). Taktéž by to mohlo souviset s bezpečnostními oprávněními, které nejsou pro uživatele vizualizace správně nastaveny (nemá oprávnění pro čtení tagů vyskytujících se ve vizualizaci). Výskyt chyby nemá žádnou periodu, ani není vztažen k určitému času. Chyba se stávala i v časech, kdy s nástroji PI System nepracoval.

9 Závěr

Rešerše vytvořená v úvodní části této práce poukazuje na aktuální témata SmartHome ve spojení s IoT. Mezi ně patří bezpečnost IoT zařízení, úspora energií, využívání chytré domácnosti k monitorování zdraví uživatelů, ale především využívání metod strojového učení na data, která SmartHome poskytuje.

V teoretické části je vytvořen popis PI System s podrobným popisem nástrojů, které byly v práci využity. Tato kapitola se až do konce řídila tím, jak byla vyvíjena praktická část práce. Součástí teoretické části je popis strojového učení, v rámci čehož je uveden podrobný popis predikčních metod náhodný les, CHAID a neuronové sítě.

Pokus o zajištění výpočtu predikce v reálném čase na cloudových platformách IBM Cloud a Microsoft Azure byl časově velmi náročný. Od začátku jsem pracoval s verzí, kdy budou data z PI Server v reálném čase streamována na cloudové úložiště IBM Cloud a v rámci cloud computing bude zajištěn výpočet predikce koncentrace CO₂. Nejprve jsem pracoval s nástrojem SPSS Modeler, ve které jsou vypracovány experimenty zaměřené na výběr a nastavení predikčních metod náhodný les, neuronové sítě a CHAID. Nejlepších výsledků v experimentech bylo dosaženo u predikčního modelu náhodný les s vytvořenými pěti regresními stromy a maximální hloubkou stromu 10. Při vytváření experimentů v cloudových nástrojích AutoAI (IBM Cloud) a Automated ML (Microsoft Azure) byly nejlépe hodnoceny modely náhodný les a ExtraTrees.

Pro vytvoření a naučení modely jsem vytvořil nasazení i v rámci cloud computingu IBM Cloud. To se však okamžitě ukázalo z finančních důvodů naprosto nepoužitelné. Poté jsem přešel na cloud Microsoft Azure. Ten se časem ukázal též jako nepoužitelný. Oproti IBM je sice mnohem levnější, přesto by byl dlouhodobý výpočet predikce drahý (zhruba 30 dolarů za týden). Tím se z cloudových platform stala slepá cesta v oblasti dlouhodobého zajištění výpočtu predikce v reálném čase.

V jazyce Python jsem vytvořil kód, který zajišťuje výpočet predikce a komunikaci s PI Web API. Prostřednictvím knihoven Scikit-Learn je naimplementována metoda strojového učení náhodný les, předzpracování dat a výpočet kritérií predikce. Konektivita dat mezi PI Serverem a osobním počítačem zajišťuje PI Web API. Pro streamování dat v reálném čase jsem uvažoval použití nástroje PI Integrator for Business Analytics. U něj se podařilo zajistit pouze instalaci, nastavení a základní odzkoušení při streamování dat do Apache Kafka a následné čtení dat ve vytvořeném konzumentovi. To však pouze v rámci serveru, na kterém je PI Integrator nainstalovaný. Jeho úplné použití pro streamování dat bránilo nastavení zabezpečení serveru, na kterém je instalován. Jeho použití a postup nasazení jsem popsal v jedné z kapitol. Z důvodu korona krize byla instalace PI System na novou budovu CPIT TL3 opožděna. Tím se protáhlo i zajištění a nastavení nástrojů PI System. Konkrétně PI Integrator byl nainstalován dva týdny před termínem odevzdání diplomové práce. V ten samý čas byl teprve zajištěn účet, jehož přihlašovací údaje jsou nezbytné pro autentifikaci při přístupu k PI Server.

Při spuštění výpočtu predikce z dat CPIT TL3 v reálném čase nastal problém s přesností predikovaných dat. Výsledky predikce byly nedostačující. Problémem byl způsob, jakým je předzpracování vstupních dat pro predikci provedeno. Bylo by třeba implementovat adaptivní normalizaci, na to už bohužel nebyl čas.

Na závěr této práce byla vytvořena vizualizace dvou bytů nacházejících se na budově CPIT TL3. Jsou v ní obsažena všechna dostupná data. Některá data, která měla být ve vizualizaci obsažena, nebylo možné dohledat. Vizualizace doplňuje stávající vizualizaci budov FEI a dřevodomku.

Výsledky této práce otevřely možnosti hlubší analýzy dat a aplikaci metod strojového učení na data PI System budov FEI, CPIT TL3 a dřevodomku. Na postupech, které jsem zde popsal a ověřil, je možné provádět predikce dat nejen koncentrace CO₂.

Možných kroků navazujících na tuto práci je více. Především je to implementace adaptivní normalizace pro zpřesnění predikce v reálném čase. Další krokem navazujícím na výsledky této práce je implementace adaptivní filtrace, která by měla výsledky predikce ještě zlepšit a měla by vyhladit průběh predikovaných dat. Také by bylo vhodné uvést do plného provozu PI Integrator for Business Analytics jak samotného, tak ve spojení s Apache Kafka.

10 Literatura

- [1] LIN, C.M. a M.T. CHEN. Design and implementation of a smart home energy saving system with active loading feature identification and power management. *IEEE* [online]. [cit. 2020-03-19]. DOI: 10.1109/IFEEC.2017.7992131. Dostupné z: <https://ieeexplore.ieee.org/document/7992131>
- [2] LI, Weixian, Thillainathan LOGENTHIRAN, Van-Tung PHAN a Wai Lok WOO. A Novel Smart Energy Theft System (SETS) for IoT-Based Smart Home. *IEEE* [online]. 06 March 2019 [cit. 2020-03-19]. DOI: 10.1109/JIOT.2019.2903281. Dostupné z: <https://ieeexplore.ieee.org/document/8661504>
- [3] AHSAN, Umar a Abdul BAIS. Distributed Smart Home Architecture for Data Handling in Smart Grid. *IEEE* [online]. 19 April 2018 [cit. 2020-03-19]. DOI: 10.1109/CJECE.2017.2776975. Dostupné z: <https://ieeexplore.ieee.org/document/8341516>
- [4] MOKHTARI, Ghassem, Amjad ANVARI-MOGHADDAM a Quig ZHANG. A New Layered Architecture for Future Big Data-Driven Smart Homes. *IEEE* [online]. 30 January 2019 [cit. 2020-03-19]. DOI: 10.1109/ACCESS.2019.2896403. Dostupné z: <https://ieeexplore.ieee.org/document/8629861>
- [5] VERMA, Prabal a Sandeep K. SOOD. Fog Assisted-IoT Enabled Patient Health Monitoring in Smart Homes. *IEEE* [online]. 07 February 2018 [cit. 2020-03-19]. DOI: 10.1109/JIOT.2018.2803201. Dostupné z: <https://ieeexplore.ieee.org/document/8283747>
- [6] LIAO, Lun-De. Design and Validation of a Multifunctional Android-Based Smart Home Control and Monitoring System. *IEEE* [online]. 31 October 2019 [cit. 2020-03-19]. DOI: 10.1109/ACCESS.2019.2950684. Dostupné z: <https://ieeexplore.ieee.org/document/8888232>
- [7] LEE, Ying-Tsung. Privacy-preserving data analytics in cloud-based smart home with community hierarchy. *IEEE* [online]. 21 August 2017 [cit. 2020-03-19]. DOI: 10.1109/TCE.2017.014777. Dostupné z: <https://ieeexplore.ieee.org/document/8013260>
- [8] HAN, Jinsoo. Smart home energy management system including renewable energy based on ZigBee and PLC. *IEEE* [online]. 14 July 2014 [cit. 2020-03-19]. DOI: 10.1109/TCE.2014.6851994. Dostupné z: <https://ieeexplore.ieee.org/document/6851994>
- [9] ZHOU, Pan a Guohui ZHONG. Privacy-Preserving and Residential Context-Aware Online Learning for IoT-Enabled Energy Saving With Big Data Support in Smart Home Environment. *IEEE* [online]. 06 March 2019 [cit. 2020-03-19]. DOI: 10.1109/JIOT.2019.2903341. Dostupné z: <https://ieeexplore.ieee.org/document/8661524>
- [10] YASSINE, Abdulsalam, Shailendra SINGH a Atif ALAMRI. Mining Human Activity Patterns From Smart Home Big Data for Health Care Applications. *IEEE* [online]. 27 June 2017 [cit. 2020-03-19]. DOI: 10.1109/ACCESS.2017.2719921. Dostupné z: <https://ieeexplore.ieee.org/document/7959184>
- [11] QOLOMANY, Basheer. Leveraging Machine Learning and Big Data for Smart Buildings: A Comprehensive Survey. *IEEE* [online]. 03 July 2019 [cit. 2020-03-19]. DOI: 10.1109/ACCESS.2019.2926642. Dostupné z: <https://ieeexplore.ieee.org/document/8754678>
- [12] SHIN, Daemin. A Security Protocol for Route Optimization in DMM-Based Smart Home IoT Networks. *IEEE* [online]. 26 September 2019 [cit. 2020-03-19]. DOI: 10.1109/ACCESS.2019.2943929. Dostupné z: <https://ieeexplore.ieee.org/document/8850082>

- [13] AL-ALI, A.R. a Imran A. ZUALKERNAN. A smart home energy management system using IoT and big data analytics approach. *IEEE* [online]. November 2017 [cit. 2020-03-19]. DOI: 10.1109/TCE.2017.015014. Dostupné z: <https://ieeexplore.ieee.org/document/8246800>
- [14] ALBERDI, Ane, Alyssa WEAKLEY a Maureen SCHMITTER-EDGEcombe. Smart Home-Based Prediction of Multidomain Symptoms Related to Alzheimer's Disease. *IEEE* [online]. 25 January 2018 [cit. 2020-03-19]. DOI: 10.1109/JBHI.2018.2798062. Dostupné z: <https://ieeexplore.ieee.org/document/8269303>
- [15] LEKIC, Milica, GARDASEVIC, Gordana. IoT sensor integration to Node-RED platform. 2018 17th International Symposium INFOTEH-JAHORINA(INFOTECH) [online]. IEEE. 2018. 1-5 [cit. 2020-16-03]. ISBN 978-1-5386-4907-7. Dostupné z <https://ieeexplore.ieee.org/document/8345544>
- [16] KOO, Jahoon, KIM, Young-Gab. Interoperability of device identification in heterogeneous IoT platforms. 2017 13th International Computer Engineering Conference (ICENCO) [online]. IEEE. 2017. 26-29 [cit. 2020-16-03]. ISSN 2475-2320. Dostupné z <https://ieeexplore.ieee.org/document/8289757>
- [17] MARKSTEINER, Stefan, JIMENEZ, Víctor, Juan, Exposito, VALIANT, Heribert, ZEINER, Herwig. An overview of wireless IoT protocol security in the smart home domain. 2017 Internet of Things Business Models, Users, and Networks [online]. IEEE. 2017. 1-8 [cit. 2020-16-03]. ISBN 978-1-5386-3197-3. Dostupné z <https://ieeexplore.ieee.org/document/8260940>
- [18] BAJER, Marcin. IoT for Smart Buildings - Long Awaited Revolution or Lean Evolution. 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud) [online]. IEEE. 2018. 149-154 [cit. 2019-16-01]. ISBN 978-1-5386-7503-8. Dostupné z <https://ieeexplore.ieee.org/document/8458006>
- [19] BASTOS, D., SHACKLETON, M., EL/MOUSSA, F. Internet of Things: A survey of technologies and security risks in smart home and city environments. *Living in the Internet of Things: Cybersecurity of the IoT – 2018* [online]. IEEE. 2018. 1-7 [cit. 2019-16-01]. ISBN 978-1-78561-843-7. Dostupné z <https://ieeexplore.ieee.org/document/8379717>
- [20] NANDI, Shubhradeep. Cloud-Based Cognitive Premise Security System Using IBM Watson and IBM Internet of Things (IoT). *Advances in Electronics, Communication and Computing* [online]. Springer. 2017. 443. 723-731 [cit. 2019-16-01]. ISBN 978-981-10-4765-7. Dostupné z DOI https://doi.org/10.1007/978-981-10-4765-7_75
- [21] TANWAR, Sudeep, PATEL, Prakash, PATEL, K., TYAGI, Sudhanshu, KUMAR, Neeraj, OBAIDAT, Mohammad, S. An advanced Internet of Thing based Security Alert System for Smart Home. 2017 International Conference on Computer, Information and Telecommunication Systems (CITS) [online]. IEEE. 2017. 25-29 [cit. 2019-16-01]. ISBN 978-1-50905957-7. Dostupné z <https://ieeexplore.ieee.org/document/8035326>
- [22] ILIEVA, Sylvia, PENCHEV, Andrey, PETROVA-ANTONOVA, Dessislava. Internet of Thing Framework for Smart Home Building. *Digital Transformation and Global Society* [online]. Springer. 2016. 450-462 [cit. 2019-16-01]. ISBN 978-3-319-49700-6. Dostupné z DOI https://doi.org/10.1007/978-3-319-49700-6_45
- [23] VASHI, Shivangi, Jyotsnamayee RAM, Janit MODI, Saurav VERMA a Chetana PRAKASH. Internet of Things (IoT): A vision, architectural elements, and security issues. In: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (ISMAC) [online]. IEEE, 2017, 2017, s. 492-496 [cit. 2017-11-07]. DOI: 10.1109/ISMAC.2017.8058399. ISBN 978-1-5090-3242-6. Dostupné z: <http://ieeexplore.ieee.org/document/8058399/>

- [24] NAZARI SHIREHJINI, Ali Asghar a Azin SEMSAR. Human interaction with IoT-based smart environments. *Multimedia Tools and Applications* [online]. 2017, 76(11), 13343-13365 [cit. 2017-11-08]. DOI: 10.1007/s11042-016-3697-3. ISSN 1380-7501. Dostupné z: <http://link.springer.com/10.1007/s11042-016-3697-3>
- [25] GE, Mengmeng, Jin B. HONG, Walter GUTTMANN a Dong Seong KIM. A framework for automating security analysis of the internet of things. *Journal of Network and Computer Applications* [online]. 2017, 83, 12-27 [cit. 2017-11-08]. DOI: 10.1016/j.jnca.2017.01.033. ISSN 10848045. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1084804517300541>
- [26] KUMAR, K. Mathan a R.S. VENKATESAN. A design approach to smart health monitoring using android mobile devices. In: 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies [online]. IEEE, 2014, 2014, s. 17401744 [cit. 2017-11-08]. DOI: 10.1109/ICACCCT.2014.7019406. ISBN 978-1-4799-3914-5. Dostupné z: <http://ieeexplore.ieee.org/document/7019406/>
- [27] *About Osisoft* [online]. [cit. 2020-02-15]. Dostupné z: <https://www.osisoft.com/about-osisoft/>
- [28] PI system. In: *Osisoft.com* [online]. [cit. 2020-02-24]. Dostupné z: <https://www.osisoft.com/pi-system/>
- [29] PI server. In: *Osisoft.com* [online]. [cit. 2020-02-24]. Dostupné z: <https://www.osisoft.com/pi-system/pi-capabilities/pi-server/>
- [30] *PI World 2020 Lab: Using PI Web API* [online]. 2020 [cit. 2021-02-21]. Dostupné z: <https://osicdn.blob.core.windows.net/learningcontent/PI%20World%202020%20and%20Beyond/PIWorld%202020%20Using%20PI%20Web%20API%20From%20Beginner%20to%20Advanced.pdf>
- [31] *PI Integrator for Business Analytics 2020* [online]. [cit. 2021-04-03]. Dostupné z: https://livelibrary.osisoft.com/LiveLibrary/content/en/bi-v6/GUID-4686F1C1-DE8C-4E74-9384-B018F1078CBD#addHistory=true&filename=GUID-44254BAB-C57F-4CF0-87A7-3825D8F1799B.xml&docid=GUID-4686F1C1-DE8C-4E74-9384-B018F1078CBD&inner_id=&tid=&query=&scope=&resource=&toc=false&eventType=lcContent.loadDocGUID-4686F1C1-DE8C-4E74-9384-B018F1078CBD
- [32] *PI World 2019 Lab: Building Displays with PI Vision 2019* [online]. [cit. 2021-4-22]. Dostupné z: <https://osicdn.blob.core.windows.net/learningcontent/pdfs/2019%20PI%20World%20Building%20Displays%20with%20PI%20Vision%202019.pdf>
- [33] PI Data Archive 3.4.390 System Management Guide. In: *Wenit.co.kr* [online]. 2014 [cit. 2020-02-24]. Dostupné z: http://www.wenit.co.kr/new_board/download.asp?board_id=pds&idx=12&fileNum=2&filename=PI_Data_Archive_2014_System_Management_Guide_EN_3_4_390.pdf Osisoft, PI-SQC_3.4.0_(2014)_User-Guide.pdf
- [34] PI ProcessBook 2014 User Guide. In: *Wenit.co.kr* [online]. 2014 [cit. 2020-02-24]. Dostupné z:

- http://www.wenit.co.kr/new_board/download.asp?board_id=pds&idx=24&fileNum=2&filename=PI%5FProcessBook%5F2014%5FUser%5FGuide%2Epdf
- [35] PI Coresight 2016 User Guide.pdf. In: *Pisquare.osisoft.com* [online]. 2016 [cit. 2020-02-24]. Dostupné z: <https://pisquare.osisoft.com/docs/DOC-2082>
- [36] *PI AF identities and mappings* [online]. [cit. 2021-02-22]. Dostupné z: https://livelibrary.osisoft.com/LiveLibrary/content/en/server-v14/GUID-AF9F91EB-E6F9-4B5C-9338-A2C8F86004F5#addHistory=true&filename=GUID-8F225C2C-6F06-41AD-80D4-74D877A4E91F.xml&docid=GUID-AF9F91EB-E6F9-4B5C-9338-A2C8F86004F5&inner_id=&tid=&query=&scope=&resource=&toc=false&eventType=lcContent.loadDocGUID-AF9F91EB-E6F9-4B5C-9338-A2C8F86004F5
- [37] *Configuring PI System Security* [online]. [cit. 2021-4-22]. Dostupné z: <https://osicdn.blob.core.windows.net/learningcontent/Online%20Course%20Workbooks/Configuring%20PI%20System%20Security.pdf>
- [38] CHOLLETA, Françoise. *Deep learning v jazyku Python*. Grada Publishing, 2018. ISBN 978-80-271-2750-4.
- [39] *Neuronové sítě, jednotlivý neuron* [online]. In: . s. 18 [cit. 2020-06-08]. Dostupné z: <https://portal.matematickabiologie.cz/res/f/neuronove-site-jednotlivy-neuron.pdf>
- [40] *Neuronové sítě*. In: *Mendelova univerzita v Brně* [online]. [cit. 2020-06-12]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471
- [41] *Koncept umělé neuronové sítě*. In: *Portal.matematickabiologie* [online]. [cit. 2020-05-01]. Dostupné z: <https://portal.matematickabiologie.cz/index.php?pg=analiza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--uvod-do-neuronovych-siti--koncept-umele-neuronove-site>
- [42] KLASCHKA, Jan a Emil KOTRČ. *KLASIFIKAČNÍ A REGRESNÍ LESY* [online]. 2004, , 8 [cit. 2020-07-28]. Dostupné z: <https://www.statspol.cz/robust/robust2004/klaschka.pdf>
- [43] KOMPRDOVÁ, Klára. *Rozhodovací stromy a lesy*. 2012. Výuková skripta. Masarykova univerzita. str. (9 – 12)
- [44] FREYVALD, Michal. *CHAID - technika konstrukce rozhodovacích stromů*. 2014. Bakalářská práce. Vysoká škola ekonomická v Praze.
- [45] CHAID node. *IBM Cloud Pak for Data* [online]. [cit. 2020-07-15]. Dostupné z: <https://dataplatform.cloud.ibm.com/docs/content/wsd/nodes/chaid.html>
- [46] MAE and RMSE — Which Metric is Better? *Medium* [online]. 2016 [cit. 2020-04-22]. Dostupné z: <https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d>
- [47] Regression — Why Mean Square Error? *Medium* [online]. 2019 [cit. 2020-04-22]. Dostupné z: <https://towardsdatascience.com/https-medium-com-chayankathuria-regression-why-mean-square-error-a8cad2a1c96f>
- [48] Mean absolute percentage error (MAPE). *Statisticshowto* [online]. [cit. 2020-07-02]. Dostupné z: <https://www.statisticshowto.com/mean-absolute-percentage-error-mape/>

- [49] *Analýza chybějících hodnot* [online]. [cit. 2021-01-15]. Dostupné z: https://www.soc.cas.cz/sites/default/files/soubory/analyza_chybejicich_hodnot_kapitola_4.pdf
- [50] Linear correlation: the linear association between variables. *Udemy* [online]. 2019 [cit. 2020-04-22]. Dostupné z: <https://blog.udemy.com/linear-correlation/>
- [51] K-násobná křížová validace. In: *Portal.matematickabiologie* [online]. 2018 [cit. 2020-04-29]. Dostupné z: <https://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--vicerozmerne-metody-pro-analyzu-dat--klasifikace--hodnoceni-uspesnosti-klasifikace--k-nasobna-krizova-validace>
- [52] *Technická univerzita Ostrava* [online]. 2020 [cit. 2021-02-09]. Dostupné z: <https://www.fei.vsb.cz/cs/pr/platforma-novych-technologii-fei-cpit-tl3/>
- [53] *Why Data Normalization is necessary for Machine Learning models* [online]. 2018 [cit. 2021-4-22]. Dostupné z: <https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029>
- [54] GUPTA, Vibhut a Rattikorn HEWETT. *Adaptive Normalization in Streaming Data* [online]. [cit. 2021-4-29]. Dostupné z: <https://arxiv.org/ftp/arxiv/papers/1910/1910.07696.pdf>
- [55] *Apply Predictive Machine Learning Models to Operations*. 2018.

11 Seznam přílohy

Příloha A: Seznam tabulek překladu použitých při tvorbě Digital States..... 85

Příloha B: Program pro predikci koncentrace CO₂ a komunikaci s PI Web API, Python

Příloha A

Tabulka B.1: Seznam provozních stavů a příslušných možných stavů pro místnosti

Provozní stav	Seznam možných stavů
Indikace energetické účinnosti místnosti	{Nedefinovana; Slaba; Dostatecna; Dobra; Vynikajici}
Indikace en. účn. osvětlení v míst.	{Nedefinovana; Slaba; Dostatecna; Dobra; Vynikajici}
Indikace en.účn. stínění v míst.	{Nedefinovana; Slaba; Dostatecna; Dobra; Vynikajici}
Indikace en. účn. stínění HVAC	{Nedefinovana; Slaba; Dostatecna; Dobra; Vynikajici}
Indikace kvality vzduchu v míst.	{Nedefinováno; Slaba; OK; Dobrý}
Stav topení/chlazení	{Nic; Topení; Chlazení}
Stav segmentu místnosti	{Normální stav; Zásah aktivní; Chyba; Alarm; Chyba a alarm}
Pracovní režim okruhu	{Vypnout; Ochrana; Úsporný režim; Prekomfort; Komfort; Předežhřátí; Ochlazení; Teplotní ochrana místnosti; Ochrana před přetečením kond.; Volné chlazení; Noční chlazení; Ventilace; Teplotní ochrana zařízení; Objemový průtok vzduchu VYP; Nepoužito; Nepoužito; Nepoužito}
Pracovní strategie osvětlení	{Energetická účinnost; Optimalizace komfortu}
Režim ovládání osvětlení pro Přítomnost	{Nic; Zohlednit přítomnost osob; Zohlednit nepřítomnost osob; Zohlednit přít. a nepřít. osob}
Centrální spínač podmínky	{Nic; Podmínka energetické účinnosti; Podmínka komfortu}
Spínač podmínky komfortu	{Nic; Automatika; Ručně}
Spínač podmínky energetické účinnosti	{Nic; Automatika; Ručně}
Podmínka ručního řízení	{Odemknout; Zamknout}
Aktuální provozní mód	{Ochrana; Úspora; Prekomfort; Komfort}
Pracovní režim místnosti	{Ochrana; Úspora; Prekomfort; Komfort}
Strategie ovládání stínění	{Energetická účinnost; Ochrana před oslněním; Posun nahoru; Posun dolů; Závislé na přítomnosti}
Stav místnosti	{Normální stav; Zásah aktivní; Chyba; Alarm; Chyba a alarm}
Okruh	{Provozní; Zařízení zastaveno; Zařízení není přirazeno; Zařízení chybi; Konfigurace zařízení; Zařízení nelze přiradit; Chybející nebo chybná konfigur.; Hledání}
Vstupní režim klimatu v míst.	{Automatika; Ochrana; Úspora; Prekomfort; Komfort}

Tabulka B.2:

Provozní stav	Seznam možných stavů
Povel	{Budova; Patro 02; Vzduchotechnická jednotka 2.1; Odtahový ventilátor; Povel}
Pracovní stav Prekomfort	{Vyp; Stupen 1; Stupen 2; Stupen 1 povolen; Stupen 2 povolen}
Pracovní stav Prekomfort	{Vyp; Stupen 1; Stupen 2; Stupen 1 povolen; Stupen 2 povolen}
Provozní mód manuálně	{Auto; Vyp; Stupen 1; Stupen 2}
Aktuální provozní mód	{Nouzové vyp.; Vyp; Stupen 1; Stupen 2; Stupen 3; Stupen 4; Zamraz; Odkour.priv.; Odkour.odtah; Odkour.privod; Nocní chlaz.; Rychlý ohrev; Rychle chl.; Vetrání; Uziv. mod}
Provozní mód manuálně	{Auto; Vyp; Zap}
Aktuální provozní mód	{Nouzové vyp; Vyp; Zap; Rezervováno; Rezervováno; Rezervováno; Zamraz; Odkour.priv.; Odkour.odtah; Odkour.privod; Nocní chlaz.; Rychlý ohrev; Rychle chl.; Vetrání; Uziv. mod}